

統計的性質に基づく文字の高精度認識に関する研究

著者	勝山 裕
学位授与機関	Tohoku University
学位授与番号	11301甲第15929号
URL	http://hdl.handle.net/10097/58706

博士學位論文

論文題目 統計的性質に基づく文字の高精度認識に
関する研究

提出者 東北大学大学院情報科学研究科
応用情報科学 専攻

学籍番号 A 9 I D 4 0 0 3

氏 名 勝 山 裕

目次

第 1 章	序論	1
1.1	文字認識による人類への貢献	1
1.1.1	文字取り込み装置の発展 (1800 年代後半)	3
1.1.2	最初の OCR の概念 (1900 年～1930 年頃)	3
1.1.3	OCR の始まり (1940 年頃～1960 年頃)	5
1.1.4	第 1 世代の OCR (1960 年頃～1965 年頃)	7
1.1.5	第 2 世代の OCR (1965 年頃～1970 年代中盤)	8
1.1.6	第 3 世代の OCR (1960 年代後半～1980 年代)	10
1.1.7	その後の OCR (1980 年代中盤～2000 年代)	11
1.1.8	最近の OCR (2000 年代以降)	12
1.2	文字認識に関連する業界の研究状況、解くべき課題と解決方法の概略 . . .	16
1.3	本論文の構成	19
第 2 章	文字認識技術の基礎知識、主要な概念	23
2.1	文字認識技術の基礎知識	23
2.2	高精度文字認識の基礎知識	26
2.2.1	類似度、距離値を使う手法	27
2.2.2	カテゴリ間の識別境界を使う手法	30
2.3	従来の高精度認識手法の課題	33
2.4	高速化の基礎知識	35
2.5	省メモリ化の基礎知識	35
2.6	従来技術の課題と本論文のアプローチ	36
第 3 章	高速化のための研究成果	39

3.1	はじめに	39
3.2	Margin Added Hashing (MAH)	46
3.3	評価実験	49
3.4	考察	53
3.5	結論と今後について	61
第 4 章	省メモリ化のための研究成果	63
4.1	はじめに	63
4.2	MQDF 手法	64
4.3	Compact MQDF 手法	66
4.4	Hybrid Compact MQDF 手法	69
4.5	評価実験	75
4.5.1	hybrid Compact MQDF 辞書のパラメータ	75
4.5.2	学習用文字	76
4.5.3	評価対象文字	76
4.5.4	比較実験	77
4.5.5	D_Q と L を固定した場合の評価	77
4.5.6	thd と L を固定した場合の評価	77
4.5.7	thd と D_Q を固定した場合の評価	78
4.6	考察	78
4.6.1	D_Q と L を固定した場合	78
4.6.2	thd と L を固定した場合	81
4.6.3	thd と D_Q を固定した場合	82
4.7	結論	83
第 5 章	作成した認識技術を使用して映像中のテロップ認識への応用結果	85
5.1	はじめに	85
5.2	提案手法	88
5.2.1	色分解 2 値画像レイヤーの作成	89
5.2.2	使用する OCR エンジンと、OCR の確信度	90
5.2.3	ノイズ除去	91
5.2.4	文字列行領域抽出	92

5.2.5	各レイヤー、各文字列行においての黒画素連結領域抽出と文字認識	93
5.2.6	キー文字矩形抽出	93
5.2.7	文字候補矩形抽出と認識	93
5.2.8	ダイナミック・プログラミング手法による文字候補矩形の選択	94
5.2.9	文字ストローク画像生成	97
5.2.10	文字認識	98
5.3	評価実験と考察	98
5.3.1	OCR エラー	99
5.3.2	色クラスタリング・エラー	100
5.3.3	DP エラー	100
5.3.4	文字候補矩形抽出エラー	100
5.4	結論と今後について	101
第 6 章	結論と今後の展望	103
	謝辞	111
	参考文献	113
	発表文献	125

目次

1.1	Tausheck の特許	4
2.1	文字認識処理フロー	23
2.2	1 文字認識処理フロー	24
2.3	加重方向ヒストグラム特徴作成フロー	27
2.4	ニューラルネットワーク	31
2.5	SVM	32
2.6	Ada Boost の学習過程	33
2.7	Ada Boost の識別過程	34
2.8	2 段階選抜方式	35
3.1	2 段階選抜方式	40
3.2	VQ クラスタリング手法の例	42
3.3	ANN 手法の例	43
3.4	LSH 手法の例	44
3.5	AM 手法の例	45
3.6	MAH 手法の辞書	47
3.7	MAH 手法の例	50
3.8	認識処理全体の処理時間と分類エラー率の関係	53
3.9	カテゴリ選択率と分類エラー率の関係	54
3.10	大分類処理だけの処理時間と分類エラー率の関係	55
3.11	VQ クラスタリング法が選択失敗して、MAH 法が成功する例	56
3.12	ANN 法が選択失敗して、MAH 法が成功する例	57
3.13	各手法での選択エラー文字数	58

3.14	AM 法が選択失敗して、MAH 法が成功する例	59
4.1	MQDF 辞書	67
4.2	Compact MQDF 法の Codebook と Indices	68
4.3	Compact MQDF 手法の辞書	70
4.4	カテゴリ代表ベクトルのクラスタリング結果	72
4.5	提案手法の辞書	73
4.6	提案手法の認識処理	75
4.7	データセット 1 の評価結果, $D_Q=4$, $L=50$ Fix.	78
4.8	データセット 2 の評価結果, $D_Q=4$, $L=50$ Fix.	79
4.9	データセット 1 の評価結果, $thd=600$, $L=50$ Fix.	80
4.10	データセット 2 の評価結果, $thd=600$, $L=50$ Fix.	81
4.11	データセット 1 の評価結果, $thd=600$, $D_Q=4$ Fix.	82
4.12	データセット 2 の評価結果, $thd=600$, $D_Q=4$ Fix.	83
5.1	SVM 手法で文字ストローク画像抽出に失敗する例	87
5.2	提案手法のフローチャート	89
5.3	入力カラーテロップ画像と色分解レイヤーの各 2 値画像	90
5.4	ノイズ除去後の各レイヤーの 2 値画像	92
5.5	各レイヤーの黒画素連結領域	93
5.6	各レイヤーのキー文字矩形	94
5.7	文字候補矩形抽出と文字認識	95
5.8	各レイヤーの文字候補矩形	96
5.9	レイヤーをまたいだ DP 処理による文字ストローク画像抽出と文字認識	97
5.10	DP 処理に使用される隣接 3 矩形の関係	97
5.11	選択された文字ストローク画像	97
5.12	色クラスタリングエラー	100

表目次

3.1	従来手法の3要件の評価比較	44
3.2	記号の意味	47
3.3	比較手法における正しく選択できた文字数	60
3.4	各手法の辞書サイズ	61
5.1	テスト1の評価結果	98
5.2	テスト2の評価結果	99
5.3	テスト2の評価結果	99

第 1 章

序論

1.1 文字認識による人類への貢献

およそ 200 万年前、アフリカ大陸の東部で類人猿から進化した現世人類は、その生命をおびやかす他の動物や自然災害から身を守るため、また、食料や水などの必需品を手に入れて生命を維持するために、グループで生活することが必要であった。そのためにグループ内でのコミュニケーションによる情報伝達は、生きるために非常に重要であったであろう [1]。

音声による情報伝達は、動物が本来持っている能力であるので、学習不要で身につけ、また、使用する（発声する）ときも簡単で素早くできる特性がある。そのため、進化前の時代を含めて、長い間、音声だけ、もしくは音声とジェスチャーだけが情報伝達の手段であった [2]。

しかし、音声の情報伝達は、その場限りで記録が残らないこと、一度に少人数しか情報伝達できないこと、また、複雑なことを表現しにくく伝えにくいという欠点があった。そのため、人口が増え、グループが大きくなるにつれて、また、脳が発達したことにより複雑な概念を考えることができるようになるにつれて、音声だけの情報伝達では不都合なことが多くなってきた。大勢の人にわかりやすく効率よく伝達するため、または、時間、場所を隔てて伝達するためには目で見ることができ、模様として記録できる情報伝達手段の発明は必須であった。

人類は、他の動物と異なり両手が自由に使用できるようになった。そのため、手を使った模様、図形は、早くから描かれていたと想像できる。この中で、あるグループの人の間では、特定の模様、図形が、特定の意味を表すように用いられ始めた。この文字の起源と

しての模様の使用は、いつから始まったのか現在では明確にはできていないが、一説には、これらが出現したのは前期新石器時代、紀元前7千年紀ごろ、または、もっと以前の可能性もあるといわれている。特にヴィンチャ文字は紀元前7千年紀の単純なシンボルに始まり、紀元前6千年紀を通して徐々に複雑さを増していき、紀元前6千年紀のタルタリアのタブレットで記号の行が注意深く整列され、「文章」の印象を与えるまでに達した進化を示している。しかし、この記号の意味はほとんど解明されていないため、正確にどの時点で原文字が文字に進化したと言うことは難しい。

しかし、このような記号の体系は、後に発達する文字に大きな影響を与えた。最初の文字体系が発明されたのは紀元前4千年紀後半の後期新石器時代に青銅器時代が始まったのとはほぼ同時期である。最初の文字体系はシュメールで発明され、紀元前3千年紀後半までにウル第三王朝時代の古代楔形文字へ発達したと一般に信じられている。同時代に、原エラム文字がエラム線文字へと発達していった。

エジプトヒエログリフの発達もメソポタミア文字と並行しているが、独立であるとは限らない。エジプトの原ヒエログリフ記号体系は紀元前3200年までに古代ヒエログリフへ発達し(ナルメルのパレット)、紀元前3千年紀までに読み書きの能力をさらに広げていった(ピラミッドの文章)。インダス文字は、紀元前3千年紀に、原文字の一形態としてか、すでに古代の様式の文字となって発達した。

漢字は紀元前16世紀頃(殷王朝初期)、中近東の文字と独立に、およそ紀元前6000年までさかのぼる後期新石器時代の中国の原文字体系から発祥したかもしれない。他にも、先コロンブス時代のアメリカ州(オルメカやマヤを含む)の文字体系も独立した起源を持つ。

このように、音声による情報伝達の不自由さを克服し、大勢の人にわかりやすく効率よく伝達するため、または、時間、場所を隔てて伝達するために、人類は文字を発明し、文明が発達した。

コンピュータが発明されて以来、情報取得の効率化の目的でコンピュータが文字を読み取る、文字認識技術のニーズが出てきた。

文字認識技術は、古くから研究され、人間とコンピュータのインターフェースとして利用されてきた。光学文字認識(Optical Character Recognition)は、活字または、手書き文書の画像をコンピュータが編集できる形式(文字コードの列)に変換する装置または、ソフトウェアである。一般にOCRと略記される。OCRは、人工知能やマシンビジョンの研究分野として始まった。OCR技術の歴史は、紙の上の文字を機械に取り込むところから始まったといえ、以下のような経緯をたどってきた。ここでは、世代別に紹介する。

1.1.1 文字取り込み装置の発展（1800 年代後半）

OCR の起源は、19 世紀後期のスキャナ機能を持った装置の発明まで遡る。1870 年に、ボストン・マサチューセッツの C.R.Carey が、光電管を寄せ集めて作った、映像電送システムである網膜スキャナを発明した [3]。1884 年に、ポーランドの Paul Gottlieb Nipkow は、現代のテレビおよび OCR の両方にとって大きなブレークスルーとなる、連続スキャナを発明した。これは、彼が発明した螺旋に穿孔されたディスク（Nipkow disk）により、画像を点の列に分解するものである。これらの発明のよって、紙の上の文字パターン情報が、電子的な情報として機械内に取り込まれることができるようになった。

1.1.2 最初の OCR の概念（1900 年～1930 年頃）

機械内に文字パターンが取り込まれるようになると、次にその情報を活用する動きがでてきた。その中の 1 つである、OCR の原点は、電信技術の拡張と、視覚障害者が文字を読むための機械の開発という 2 つの問題にまつわる活動である [4]。1914 年、Emanuel Goldberg は、文字列を読み取り電信符号に変換する機械を開発した。同じころ Edmund Fournier d'Albe は、オプトフォンという携帯型スキャナを開発した。これは、この機械を印刷物のページ上ですべらせると、文字の形状によって異なる音を発するものである。

人々は、人間の代わりに機械が文字を読めないか、と考えるようになり、OCR のニーズが高まった。Emanuel Goldberg は、データ入力手段として OCR 技術の開発を続けた。後に彼は、画像を光電管で読み取り、必要な識別パターンを含むテンプレートと比較することでマッチングを行うという技法を提案した。

1929 年、Gustav Tauschek も同様のアイデアを思いつき、ドイツで OCR に関する特許を取得した。アメリカでは、1933 年、General Electric Company の Paul W. Handel が同様のテンプレート・マッチング方式の OCR 技術の特許を取得している（アメリカ合衆国特許第 1,915,993 号）[5]。1935 年、Gustav Tauschek もアメリカで特許を取得した（アメリカ合衆国特許第 2,026,329 号）[6]。代表として、Tauschek の特許を紹介する。図 1.1 は、この特許の詳細な構造を示す。このマシンでは、黒インキで数字が印刷された紙 1 は、レンズ 3 の正面にくるように板 2 の上に置かれる。数字は、光源 4 からの強い光によって照らされる。レンズ 3 を通した反対側では、光電子セル 5 が、最小のイルミネーションに反応するように、適切な距離に配置される。

0～9 までの数字に対応するステンシル（文字型を切り抜いた型紙）をもつ比較ホイー

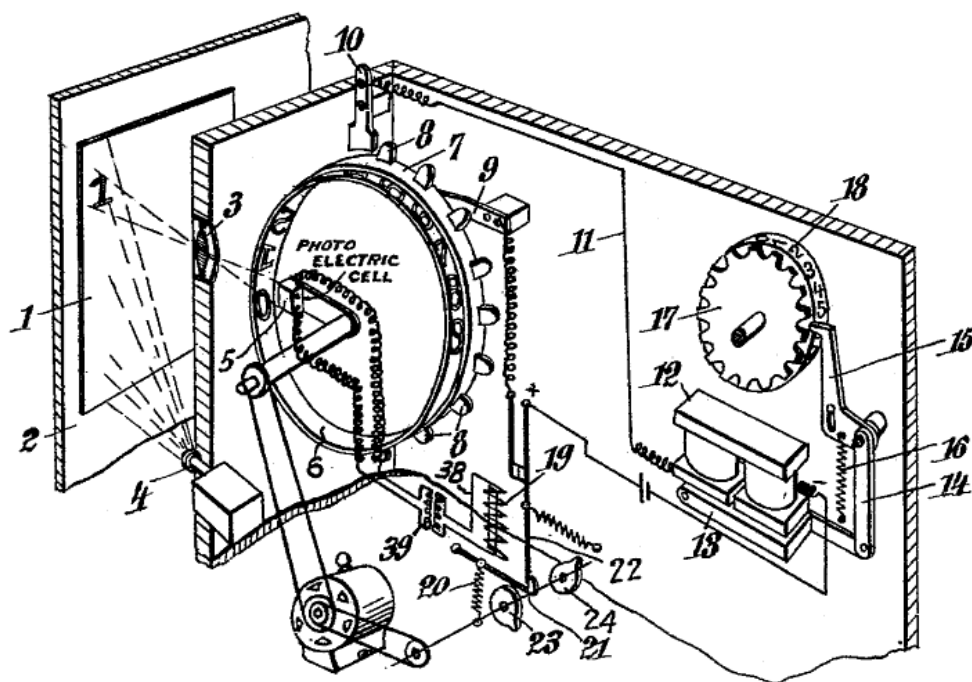


図 1.1 Tauscheck の特許

ル 6 は、図のように設置される。ホイール 6 が回転し、ステンシルがセル 5 の前にきた時には、紙 1 上の数字から発した光線は、ステンシルを通してセル 5 に到達する。もし、数字が印刷されていない白紙が置かれたなら、ステンシルを通過した光は、数字の完全な形となって光電子セル 5 を照らす。この時、光電子セルは、そのステンシルの形では最も大きな反応を示す。黒で印刷された数字が置かれた場合、その文字から発した光は、単なる白紙の場合の光の量よりも小さくなる。一般には、文字の形とステンシルの形が異なるので、文字の周囲の白紙部分からでた光の一部が、ステンシルを通過し、光電子セルに達し、光電子セルは小さい反応を示す。しかし、文字の形とステンシルの形が完全に一致すると、黒文字の背景部分の白紙から発した光は、全てステンシルに阻まれてステンシルを通過することができなくなる。そのため、光電子セルの反応は極端に小さくなる。例えば、図に示されているように、紙上の文字が「1」で、ステンシルが「2」の場合は、紙上の「1」の黒領域の周囲にある背景の白紙面から発した光は、ステンシルの 2 の形の隙間のほとんどで通過する。しかし、紙上の文字が「1」で、ステンシルも「1」の場合は、ステンシルの数字の形と紙上の数字の形が完全に一致するため、紙上の「1」の黒領域の周囲にある背景の白紙面から発した光は、ほとんど全てがステンシルに阻まれ、光電子セ

ルに届かず、反応が最小になる。この結果は、光電子セル 5 が出す信号としてリレー回路を通して出力され、機械的なセクションを駆動して、ホイール 18 を検知した数字を表示する適切な位置まで回転させることで表示される。

このように、文字を読み取るという意味での OCR の原初は、テンプレートマッチングでなされた。Tausheck の読み取りマシンによって利用されたテンプレート照合メソッドは、かなり標準の画像処理手法で、簡単だが、テンプレート照合には、いくつかの明白な制限がある。1 個のテンプレートは、同じサイズと位置の文字を見分けることができるだけで、テンプレート照合は、同じカテゴリの文字間で起きる雑音や、文字の小さい変化に非常に影響を受け易い。また、テンプレートマッチングをコンピュータ上の画像処理で行う場合、入力文字をテンプレートのすべてと画素単位で比較するために、多くの計算パワーと時間を必要とする課題がある。

1.1.3 OCR の始まり（1940 年頃～1960 年頃）

1950 年までに、技術革新が早急に進み、電子的データ処理は重要な分野になっていた。当時、データ入力パンチカードによって行なわれていた。また、増加するデータを処理するコスト効率の良い方法が必要だった。同時に、機械による文字読み取り技術のニーズが高まった。この機運から、1950 年代半ばに、OCR 機械が市販されるようになった。

1949 年、米国復員軍人援護局から視覚障害者支援のためという要請を受け、RCA の技術者らが、初期のコンピュータを使った OCR を研究した。この研究には、単に活字をコンピュータ内の文字コードに変換するだけでなく、その文字を読み上げるという部分も含まれており、初期の音声合成の研究でもある。しかし、コストがかかりすぎることが判明し、研究は挫折した [7]。1950 年、AFSA（アメリカ国家安全保障局の前身）の暗号解読者 David H. Shepard は、局の手続きの自動化の勧告案作成に取り組んだ。その中には印刷された文書をコンピュータが処理できる形式に変換する問題も含まれていた。David H. Shepard はそのようなことをする機械「Gismo」を作成することを決め、友人の Harvey Cook と共に試作に取り組んだ [8]。

1951 年に最初の商業コンピュータ UNIVAC I が稼働した後、コンピュータを使用した様々な手法が提案されていく。

1953 年、David H. Shepard は特許（アメリカ合衆国特許第 2,663,758 号）を取得 [9]。Gismo は、英語のアルファベットのうち 23 文字を読み取ることができ、モールス符号を

理解し、楽譜を読みとることができ、活字のページを読み上げることができ、タイプされたページを読みとってプリンターで複製することができた。David H. Shepard はその後 Intelligent Machines Research Corporation(IMR) を設立し、世界初のいくつかの商用 OCR システムを出荷した。Gismo も IMR のシステムも単純な文字マッチングではない画像解析をしていて、いくつかの書体を認識することができた。Gismo は画像中の文字の位置を正確に合わせる必要があったが、IMR システムではスキャン領域のどこの文字であっても認識でき、実用に耐えるものであった。

最初の商用システムは 1954 年にリーダーズ・ダイジェスト社に納入され、販売報告書をコンピュータに入力するのに使われた。これは、タイプされた報告書を読み取り、パンチカードに変換し、コンピュータに入力するもので、年間 1500 万から 2000 万部を売り上げている同社の事務処理を効率化した [4]。このシステムは後にスミソニアン博物館に無償で提供され展示されている。2 台目のシステムはスタンダード・オイルがカリフォルニア州でクレジットカードの文字を読み取るために使い、他の石油会社もこれに追随した。IMR が 1950 年代後半に販売した他のシステムとしては電話会社の請求書読み取り装置やアメリカ空軍のテレタイプ用ページスキャナーなどがある。IBM などは後に David H. Shepard から OCR 特許のライセンス供与を受けている。

また、この時代、1955 年に米ファーリントン社が、コンピュータを使用した OCR を発表した [10]。1956 年、Kelner と Glauberman がコンピュータ上でのテンプレートマッチングを開発した。これは、magnetic shift register というアナログ手法でスリット内を光検出器で観測し、スリット内の黒領域の値を獲得し、X 軸へ投影することで 2 次元情報を 1 次元に変換する。その後で、データをアナログ値からデジタル値に変換した後に、コンピュータ上でテンプレートマッチングを行う手法であった [11]。

また、1957 年に、イギリスの Solatron Electronics Group Ltd. が、peephole 法に基づいた最初の OCR で、ERA(Electric Reading Automation) と呼ばれたコンピュータを使用した OCR を発表した [12]。peephole 法は、日本語で定点観測法とよばれている。この手法は画像全体のテンプレートマッチングではなく、予め決めておいた点の位置でのテンプレートマッチングを行う物である。2 値画像の上に全てのカテゴリで共通の固定点をいくつか設定する。画像上の 7 点の画素値（白または黒画素）の列を辞書内の同じ位置の画素値のテンプレートと比較して、一致する辞書内テンプレートを認識結果とするものである。

日本では、1958 年に ETL（電総研）の飯島らが、同じ peephole スキームに基づいた装置

だが、3 レベル論理を使った E R A と比較してよりシステマチックであり、より効率的であった装置を設計した [13]。

また、この時代、磁気インクを使用した OCR (MICR) が、1957 年 GE 社で開発され、Bank of America に納入され実用された。また、日本でも 1959 年から電気試験所（現・産業技術総合研究所）で本格的な OCR の実験が開始された。

研究では、1957 年、米ベル研究所が手書き文字オンライン認識の実験を行った [10]。また、構造解析アプローチの研究として、1954 年 Rohland が、縦走査線中の黒画素領域の数を特徴とした cross counting 法を開発した。1956 年 Johnson [14] と Dimond [15] が、snode（ゾンデ法）法による特徴量抽出を開発した。これは、画像の定点にノードをおいて、その間に黒画素と交差する数を特徴量としたものである。このゾンデ法は、Kamensky [16]、Glucksman [17] により改良された。1959 年 Doyle が、基準線から黒画素までの距離を geometric 特徴として使用した。

1.1.4 第 1 世代の OCR（1960 年頃～1965 年頃）

1960 年から 1965 年までの時期に現われる商用 OCR システムは、OCR の第 1 世代と呼ばれる。この時期は試行錯誤の時代である。

第 1 世代を特徴付けるものは、OCR が読み取るために設計された制限された文字の形である。実際、この文字形の使用は、効果的だった。最も典型的なものは 1962 年に登場した NCR420 である [18]。これは、NOF または bicodes と呼ばれる、数字と 5 つの記号のための特別なフォントを使っていた。次の典型的なものは、ファーリントン Electronics 株式会社のファーリントン 3010 だった [19]。仕組みは、NCR の OCR と同じで、Selfchek12F,7B と呼ばれる、特別なフォントが使用された。また、OCR がコンピュータのための非常に重要な入力装置であると認めて、IBM も OCR の開発に非常に活動的だった。IBM で最初の商品化された OCR は IBM1418 だった [20] [21]。この装置は、特別な IBM フォント 407 を読むように設計された。この字形は前記の 2 つのフォントより自然に見えた。認識手法は論理的なテンプレート照合だったが、論理テンプレートの設計は、変化に対処するためにより複雑だった。

日本では、1960 年代の終わりまでには、いくつかのメインフレーム会社がそれらの OCR の第一世代のものを発表した。たとえば、富士通の Facom 6300A や、日立の H-852 である。両方ともストローク分析法を使用する。他方、NEC の N240D1 は、IBM1417 と

類似していて、407 フォント文字を対象文字とした。

また、1964 年に、米 REI(Recognition Equipment Inc.) 社が、マルチフォント OCR 「RETINA」を発表した。これは、ロサンゼルスデータ会社のサービス部門で使用されたが、大変高価な装置だった [22]。

研究では、以下の様な物がある。2 次元テンプレートマッチングの位置ずれに対する弱さを克服するために、特徴量に工夫が加えられた。1958 年に IBM7090 が開発された後、IBM の Horowitz と Shelton [23] は、1961 年に自己相関手法で特徴記述した OCR を研究した。また、1961 年に日本電波研究所の Sato 等は、自己相関のシミュレーションを実現した。1960 年に米コーネル大学でパーセプトロンモデルが発表された。また、飯島がテンプレートマッチングの中で正規化の重要性を明確にし、これに関したパターン認識の基礎研究を行った。これは、1962 年のパターンの正規化の理論 [24] と、1963 年の特徴抽出の理論 [25] である。ぼけの概念は、彼の仕事によって最初にパターン認識研究に導入され、Marr [26] によって西洋に広く伝えられた。1960 年代早期に、フーリエ展開をパターン認識に応用した研究が Cogriff [27] によって開始された。構造解析アプローチの研究では、1961 年 Weeks [28] が、cross counting 法を発展させた。これは、1 方向あたり 6 走査線で、1 文字を 4 方向で走査して黒画素と交差する数を特徴量としたものである。1963 年 Sakai ら [29] の asynchronous reading 法によってスリット／ストローク解析手法の一般化の研究が行われた。また、1967 年に Hart が最近傍識別理論を発表した。

1.1.5 第 2 世代の OCR (1965 年頃～1970 年代中盤)

第 2 世代の文字読み取り機械は、1960 年代中頃から 1970 年代初めに現われた。これらのシステムは、通常の印刷文字や手書き文字を認識することができた。しかし、手書きの場合は、文字セットは数字や、少数の文字、記号に限定された。この時代、特に郵便番号読み取り装置の開発が進んだ。通常印刷文字では、(ユーザによるフォント指定はあったが) マルチフォントの英数記号が認識できた。

第 2 世代 OCR を特徴つけるものとして、手書き文字と同様に普通の印刷文字の認識能力がある。この世代で、最初で有名なシステムは、IBM 1287 である。これは、1965 年にニューヨークの世界博覧会で展示された [30]。1964 年、RETINA [31] と呼ばれる非常に強力な OCR システムが、Recognition Equipment 株式会社によって開発された。このシステムは、手書き文字と印刷文字の両方を読むことができた。手書き文字の場合は、文字

は数字と、C、S、T、X、Z、+、-、およびシンボルだけに限定された。他方、印刷文字では、40種の文字、4つの特別な記号、大文字アルファベット、数字、を読むことができた。ただし、ユーザは、マシンによって読まれるために字体を指定した。RETINAは、非常に高価であったのにもかかわらず実際に使用された。

1965年ごろ、リーダーズ・ダイジェストとRCAは、同誌の広告にのせたクーポンのシリアル番号を読み取るOCR装置を共同で開発した。シリアル番号は、RCAのプリンターでOCR-Aフォントで印字された。読み取り装置はRCA 301というコンピュータに直接接続された。この技術はTWAで航空チケットの読み取りにも使われることとなる。読み取り装置は毎分1,500枚の文書処理でき、正しく読み取れなかった文書は弾かれた。RCAはこれを製品化し、保険会社などが採用した。

また、第2世代の最も実用的なシステムは、1960年代後半に実用化された郵便番号認識機械であろう。アメリカ合衆国郵便公社は、1965年から発明家Jacob Rabinowの開発した技術を元にしてOCR装置を使っている。ヨーロッパでOCRを最初に採用したのはイギリスの郵便局だった。イギリスでは1965年、郵便貯金に相当するNational GirobankがOCRを使った自動化を行った。カナダの郵便局は1971年からOCRを使用している。OCRは受取人の名前と住所を読み取って分類するために使われた。

日本では、1966年、東芝が手書き数字OCRによる郵便番号用の最初の自動的な文字選別機を開発した[32]。同年、日立は、高機能で低コストな最初のOCR機械を作り、また、日本電気もOCR市場に参画した。それらの装置は、典型的な構造解析法を使用していた。日本では、1968年7月1日に郵便番号が導入され、手書きの数字である郵便番号をOCRシステムで読み取るシステムが本格稼働した。

また、この時期に、標準化分野で、重要な仕事が行われた。1966年には、OCR必要条件に関する研究が完了し、アメリカの標準OCR文字セットOCR-Aが定義された。このフォントは高度に型にはめられ、人間には読みにくい、機械での認識を促進するものだった。欧文フォントも設計され、より自然で人間に読みやすいフォントであるOCR-Bが設計された。これらは、後年、1976年にISOで標準化され、その後、両方のフォントを読むことができる機械が現われた。

この時代の研究では、以下の様な物がある。1965年に電気試験所の飯島泰蔵が、視覚でとらえられる映像にボケが伴う事実を解明し、視覚パターンの認識に関する統一的な基礎理論体系を世界に先駆けて確立した。また、1966年から始まった通商産業省大型プロジェクト「超高性能電子計算機の研究開発」において、人間に判読できる印刷文字であれ

ば正確に読み取ることができる装置を目指して ASPET/71 の開発が行われ、1971 年に完成した。本装置では、上の統一的な基礎理論体系と、新たに飯島によって発明された「複合類似度法」が採用され、それまで不可能とされていた低品質文字の読み取りが、初めて可能になった。また、アナログ量を並列的に演算する空間回路網が開発され、上記方式による演算の高速処理を可能にし、毎秒 2,000 字の高速読み取りを実現した。1965 年に Watanabe や、1972 年の福永らによって KL 展開を文字認識に応用する研究が行われた [33] [34]。1971 年に野口、飯島は、この手法で印刷カタカナデータで実験を行った。1967 年から 1972 年にかけて、部分空間法 [35] [36] が発表された。また、テンプレートマッチングと構造解析法の中間の手法として、1968 年、NEC が「画像を縦横に分割し、各分割ブロック毎に 4 方向の方向特徴を求める特徴量」の技術を開発し、特許をイギリスで取得した [37]。類似アイデアは、Munson [38] も提案し、有名な OCR である IBM1287 でも Greanias ら [30] により提案された。1966 年に米 IBM 社は、印刷漢字 OCR を試作した。日本でも、1970 年に郵政省電波研が、手書き漢字 OCR の実験を開始した。1973 年に日立が印刷漢字 OCR 実験を開始するなど、1973 年には、国内で手書き漢字、印刷漢字の研究が本格化した。このように、1970 年代になると、手書きの英数字にカタカナまで読めるものが出現した

1.1.6 第 3 世代の OCR（1960 年代後半～1980 年代）

この世代の OCR システムでは、低品質印刷文字や手書き文字が研究対象となった。また、ハードウェアの劇的な進歩による低コスト化と高機能の追求も重要な目的であった。OCR 製品では、より精巧な OCR 製品が市場に現れ始めた。しかし、主流は、一定の文字間隔で印刷された文字や、少ない種類のフォントの使用を前提とした、より単純な OCR 製品であった。これは、普通のタイプライターでラフに作成した原稿を最終編集のために OCR 装置を通してコンピュータに入力する目的で使用された。

この頃の低品質文字を対象とした OCR は、以下のものがある。1968 年、低品質文字も対象とする OCR として、IBM が IBM1975 を発表した。また、1971 年に東芝が ASPET/71 を発表した。また、手書き文字を対象とした OCR は、以下のものがある。1975 年、日本労働省が手書きカタカナ文字 OCR の使用を発表し、1980 年に NTT システムが採用される。1983 年、東芝は、複合類似度法に基づく漢字 OCR OCR-V595 を発表。

また、1980 年に通産省パターン処理大型プロジェクトにおいて、マルチフォント OCR

を発表した。また、その後の日本語の手書き OCR 技術の発展に大きく貢献したのが、電総研が 1984 年に作成、公開した漢字文字データベースの ETL9B の存在である。同年、電総研は、ETL9B において 84% の認識精度を達成するが、このデータベースの公開によって、開発した技術が同じ評価基準で行えるようになり、評価に客観性を持たせることができるようになり、その後の技術開発による高精度化が一気に進んだ。

また、この時期は、以下のような研究がなされた。1980 年に通産省パターン処理大型プロジェクトにおいて、マルチフォント OCR が発表された。1981 年に、Yasuda が 1 つのカテゴリに属する多数のテンプレートを用意することになる、特徴空間でぼけたテンプレートを使用する方法の研究を発表 [39]。1982 年から 1988 年にかけて、Yamashita 等によりテンプレートマッチングの位置ずれ対策として、非線形正規化が盛んに研究された [40] [41] [42]。また、字種の拡大、自体の自由化、書体の自由化で研究が行われた [43] [44] [45]。手書き漢字認識では、文字ストロークを抽出してモデルと比較する構造解析法が用いられた [44] [46] [47] [48]。しかし構造解析法は雑音に弱く安定したストロークを抽出できないという課題があり、予め用意したパターンと入力データをマッチングする方法であるパターン整合法が大きな成果を得た [49] ことで主流となった。パターン整合法の OCR では、文字認識処理は 2 値化、正規化、特徴抽出、パターンマッチング、後処理のモジュール構成となる。課題は、字種が多く計算時間がかかること、予めパターン全部用意することが困難であることであった。処理時間については、階層的な分類により解決された [50]。後者の課題については、図形特徴 [51] [52]、統計的特徴 [53] の大きく 2 つの特徴抽出方法が研究開発され、印刷漢字認識に関しては実用化課題が解決された。

1.1.7 その後の OCR（1980 年代中盤～2000 年代）

この年代では、ハードウェアがより安くなり、また、OCR システムはパッケージ・ソフトとして利用可能になり始め、OCR は一般的に使用されるようになった。

OCR 機械は既に 1950 年代から市販されるようになっていたが、まだ一部でしか使用されていなかった。この主な理由は、OCR システムのコストだった。しかし、1980 年代ごろからハードウェアがより安くなり、また、OCR システムはパッケージ・ソフトとして利用可能になり始めるとともに、OCR のコストは、年々低下していき、OCR は一般に普及するようになった。

この時代、以下のような活動があった。1980 年代後半に、Caere 社は OCR ソフトウェ

ア OmniPage を販売した。1988 年、富士電機がマルチフォント活字漢字 OCR XP-70S を発表 [54]。三洋電機が 8086 マイクロプロセッサを使用した漢字 OCR CLL-2000 を発売 [55]。1990 年代中ごろ、アメリカ合衆国エネルギー省 (DOE) は情報科学研究所 (ISRI) に印刷文書の認識技術育成という使命を与えた。それにより 5 年間に渡って Annual Test of OCR Accuracy がまとめられた [56]。これにより、欧米文字の OCR 技術が発展し、その結果、近年では、欧米文字の正確な認識はほとんど解決済みの問題となった。

近年は、OCR ソフトウェアは、P C のソフトウェア製品として一般に使用されている。欧米では、ABBYY, Scansoft 社が高精度で、多言語に対応した OCR ソフトウェアを販売している。また、日本では、Media Drive 社などが日本語に対応した OCR ソフトウェアを販売している。しかし、これらの製品でも文字を正しく認識する確率は 100 % ではなく、間違いの許されない状況では人間が結果を確認する必要がある。

19 世紀および 20 世紀初頭の新聞を使った研究によると、単純に文字単位で認識する市販の OCR ソフトウェアの識字率は 71% から 98% だった [57]。手書き文字、特に筆記体の手書き文字認識や文字数の多い言語の文字認識ではまだ研究の余地がある。

文字認識の精度はいくつかの測定法で表され、実際に使用した測定法によって精度は大きく左右される。例えば、文脈や辞書を使わずに純粋に文字単位で認識する場合、識字率が 99% であっても、単語ベースの誤り率は 5% となるかもしれない [58]。

また、この時期は、以下のような研究がなされた。1987 年 部分空間法を Fukunaga が発表した。現在では、印刷漢字認識はデジタルカメラ入力を前提とした低解像度文字への対応、映像中や情景画像中の文字の認識など、撮影環境条件の緩和を課題として研究されている [59] [60]。

マッチング手法としては、認識辞書において、カテゴリの学習パターンの分布から求めた特徴量をカテゴリの代表とする方法から、多数の学習サンプルを使って、カテゴリ間の境界を詳細に記述する SVM [61] (1995 年) や Boosting [62] (1995 年) のような手法が考案されてきた。また、動物の神経回路網の働きを模倣するニューラルネットワーク手法 [63] も使われるようになった。また、テンプレートマッチングとして HMM を 2 次元に適応した研究 [64] も発表された。

1.1.8 最近の OCR (2000 年代以降)

最近の OCR 製品として、IT 各社は、OCR 機能を取り入れた製品、ネットワークまたはクラウド上のサービスを提供している。対象も名刺、写真の中の文字などを扱うものが

でできた。以下、主な物を列挙する。

米 Microsoft 社は、Windows での文書検索の一環として、Microsoft Office Document Imaging という機能を提供している。この技術は、Microsoft Office 製品の中で Tiff などの特定の形式の画像を文字認識して、認識結果テキストを画像に付与する。この結果、Office 文書と同等に画像を検索できるようになるものである。

米 Evernote 社は、クラウド上のサービスである Evernote を提供している。Evernote では、テキストだけではなく写真や画像、音声など、複数の形式のデータを 1 つのメモ（ノートと呼ばれる）にまとめて保存することができる。データ形式ごとに別々の場所に保存する必要がないので、あとでメモを探したときに関連のあるデータを見つけやすい。また、一旦保存したデータは、内部で処理されるので、メモを取りたい対象に応じて形式を変えて表示することができる。このサービス機能の 1 つに OCR があり、カメラ撮影された画像をクラウドに送信すると、クラウド上で OCR が実行されて、画像にキーワードが付与されて、画像の検索が可能になる。認識結果テキストはダウンロードして再利用も可能である。

米 Google 社は、オフィスツール「Google ドキュメント」を提供している。このツールには、画像ファイルや PDF ファイル内にある文字を抽出し、テキストに変換できる OCR 機能がある。画像や画像 PDF 中の文字をクラウドで認識して、作成したキーワードを画像や PDF に付加した Google ドキュメント形式の文書をクラウドに保存する。これにより、画像や PDF が、キーワード検索可能となる。Google ドキュメントは、2010 年 6 月から画像や PDF からアルファベットなどを読み取ってテキスト化する OCR 機能を導入していたが、2011 年 3 月から日本語を含む合計 34 の言語に対応した。

また、日本でも NTT ドコモが、文字認識をクラウド上で行うサービス「文字認識 API」を公開している。これは、スマートフォンなどで撮影した情景画像をクラウドに送信するだけで、文字認識を行い、クラウド内の単語データベースと認識結果を照合し、文字以外の不要な部分を除去しつつ誤った結果の訂正を行うことで、精度の高い文字認識を実現する技術である。

最近の研究では、印刷文字認識はデジタルカメラ入力を前提とした低解像度文字への対応、映像中や情景画像中の文字の認識など、撮影環境条件の緩和を課題として研究されている [59] [60] [65]。これらを分類すると、以下に分類できる。

- ・カメラベースの文書認識技術の研究 [66] [67]
- ・カメラベースの情景画像中の文字認識の研究 [68] [69] [70] [71] [72]

・映像中のテロップ文字認識の研究 [73] [74]

印刷文書をスキャナで取り込んだ画像と異なり、これらの研究では、認識対象の画像は、撮影対象や撮影方法、条件によって、低解像度、歪み、オクルージョン、複雑な背景、不均質な照明、フォーカスのずれ、手ぶれ、カメラ自体の特性による画質の不均一性などの課題があり、それらに対応するために、歪み補正、2 値化、高解像度化等の画像処理の研究や、カメラ撮影画像用の文字認識技術の研究が行われている。

また、スキャナなどで獲得された文書画像を対象とした研究では、従来より画質が悪い状態を対象とした研究が行われている。すなわち、認識結果から劣化状態をフィードバックさせ再認識したり [75]、入力画像の画素の分布状況から劣化程度を推定し、適応的に補正を行う手法などの劣化やノイズが付加した画像での文字認識の研究である [76]。その他には、カラー文書まで対象を広げた研究も行われている [77] [78] [79]。

手書き文字認識に関する研究では、より自由な書体に向けて、文字の変形の吸収の新しいアプローチとして 2 次元 DP によるマッチング [80] や、サインの照合 [81] [82]、郵便物の住所認識 [83] [84]、小切手処理 [85] などの研究が行われている。また、保管、閲覧が難しい歴史的な文書の管理上のニーズから歴史的な文書認識に関する研究も行われている [86] [87]。

文字認識コア部分のマッチング部分の研究としては、教師無しニューラルネットワークとして、Deep Learning が 2000 年代半ばから研究されてきた。特に、2010 年に Ciresan ら [63] により、深い非線形ニューラルネットワークをバックプロパゲーションにより学習を行ったニューラルネットワークが、MNIST の手書き文字認識において、既存の技術を凌駕したことにより一気に注目された。近年は、強力なハードウェアを使用するこの技術の研究が進行中である。

この他に、数式認識 [88] や楽譜の認識 [89]、OCR 技術の多言語への拡張の研究 [90] [91] [92] も盛んに行われている。

一方、近年、ICT 技術が広まって、コンピュータをコミュニケーションのツールとして使用することにより生活が便利になってきた。ICT 技術は、主に人間がコンピュータ内に作成した電子情報を用いて非常に効率的な情報取得、情報処理をおこなっている。例えば、Web 検索エンジンにキーワードを入力し、検索を実行すると、世界中の Web ページからキーワードに関係ある順番で Web ページがランク付けられて表示される。ユーザはそこから関係ありそうなものを選択し、さらに詳細な Web ページの内容を表示させみることができる。

しかし ICT 技術でカバーできない分野がある。その 1 つが実世界の文字の読み取りである文字認識技術の分野である。

OCR 技術は、現在、様々な分野で利用されている。古くから実用化されている郵便番号読み取り分類機械の他にも、帳票上の手書き数字を読み取りシステムへ入力するデータエントリー機能も一般に広く使用されている。また、新聞、雑誌などの文書上の活字文字を対象として、印刷されているテキストの再利用のための入力効率化や、新聞、雑誌などの文書をスキャンして画像または PDF で文書管理システムに保管しておき、後にキーワード検索するためにも使われている。この検索用途の文字認識対象は、文書画像だけでなく、映像中のテロップを対象としたものや、周囲の風景をカメラで撮影した情景画像を対象とした物など、幅広いニーズがある。

しかし、現在の OCR 技術では、読み取り精度が不十分であるために、誤認識により上に列挙した各種の機能、サービスが完全に行われない。例えば、郵便番号では一定の割合で誤分類が発生し、誤配や遅配の原因になっている。また帳票の文字入力やテキスト再利用では、人間の確認と修正が必須となっている。また、文書、映像、画像の検索では、誤認識により正しいキーワードが付与されないために、それらにあるキーワードで検索したのにもかかわらず、検索できない場合がある。

現在の OCR 技術を改良し、実用的な処理時間内で認識精度を向上させることができれば、上の様な不具合が改善される。しかし、ICT 技術は、本質的に、コンピュータの中に入った電子情報の処理であるため、上の様な実世界の文字をコンピュータに取り込む文字認識技術の改善には貢献していない。そのため、ICT 技術が発達した現在でもなお文字認識技術には課題が残っている。

これまでの文字認識技術の研究では、ハードウェアの制約をあまり考慮しないで高精度、高速性を目指した研究が主流であった。例えば、学習サンプル文字の分布を記述した辞書を使用した部分空間法、MQDF 認識処理、SVM 法、Boosting 法は、複雑な演算が必要であるため、高速処理が難しい課題がある。また、これらの手法の中には、辞書サイズが大きくなり、スマートフォンなどの小型端末で動作させることが難しかったり、または、並列に多数の処理を行う場合にハードウェア資源が大量に必要になったりするものがある。この様に、従来は、実用的な文字認識技術の研究では不十分であった。ここで、実用的な文字認識技術とは、高精度を維持したままの高速処理、かつ、コンピュータのメモリサイズの制限がある状況での高精度認識処理である。

本研究の目的は、従来の研究では不十分であった実用的な環境での高精度な文字認識技

術について新しい手法を提案し、従来実現できなかった分野で文字認識技術を使う事により、実世界の文字をコンピュータに取り込み易くして、ICT によるサービスをさらに高度化することにより人類の幸福に貢献することである。

具体的には、高速化については、認識処理過程に 2 段階選抜を導入して、高速化を行う。2 段階選抜手法自体は昔から存在するが、本研究では、2 段階選抜の 1 段目に改良を加え、1 段目に必要な 3 要素を同時に満たす新しい手法を考案する。また、高精度を維持したままの辞書サイズ圧縮では、MQDF 手法において、クラスタリングによる近似を使う新しい手法を提案する。また、従来不十分であった映像中のテロップ文字認識に提案した技術を応用した例を示し、本技術が実用されることを示す。

1.2 文字認識に関連する業界の研究状況、解くべき課題と解決方法の概略

文字認識の基本的な概念は、以下のようなものである。

文字認識を行うためには、まず読み取りたい文字から辞書を作成する必要がある。認識対象とする文字カテゴリについて、多数の学習用の文字画像（文字パターン）を収集し、それらを使用して認識用の辞書を予め作成しておく。辞書は、以下の様に作成される。

文字画像を対象に、一定の特徴抽出処理により特徴ベクトルを抽出する。カテゴリ毎に特徴ベクトルの平均値など、カテゴリを代表する特徴ベクトルを求め、カテゴリを示す文字コードとセットにして辞書とする。

認識時には、先ず入力された画像から文字がある領域を特定する。次に、個々の領域内の文字画像を対象に、辞書作成と同じ処理によって特徴ベクトルに変換する。この特徴ベクトルと、辞書の各カテゴリの代表特徴ベクトルと比較を行い、入力文字画像の特徴ベクトルに最も類似している辞書内のカテゴリの文字コードを認識結果とする。

従来の文字認識技術は、主に高精度化を目指して研究されてきた。高精度認識を実現する手法の例としては、MQDF 手法、部分空間法、ニューラルネットワーク手法、SVM 法、Boosting 手法などがある。MQDF 手法、部分空間法は、学習文字画像を多数収集して、1 つのカテゴリの文字集合の特徴空間の中での分布情報を求めそれをマッチングの時に使用するものである。また、ニューラルネットワーク手法、SVM 法、Boosting 手法は、学習文字画像を多数収集して、2 つのカテゴリ集合間の識別境界を精密に求め、マッチングの時に使用するものである。

これらの手法は、各カテゴリに代表ベクトルを1つ持ち、入力文字との比較を行う単純な手法に比べて複雑な計算が必要となる。例えば、部分空間法では、各カテゴリの学習文字画像を用いてそのカテゴリの分布を最もよく記述できる直交軸を辞書として求める。入力文字の認識処理では、特徴空間上で1点となる入力文字の特徴ベクトルを、各カテゴリの直交軸へ正射影した時の射影長を求め、カテゴリ間で最も長い射影長となるカテゴリを認識結果としている。

このように、従来の手法は、複雑な計算によって高い精度を達成していた。その反面、精度以外の、処理速度や、使用するハードウェア資源などの文字認識技術の実用上問題となる項目についてはあまり研究が進んでいなかった。ここで、実用的な文字認識とは、高精度を維持しながら、高速で、メモリ消費量が小さいものである。この観点で、従来の高精度認識手法は、以下の様な課題があった。

- ・高速化の課題

日本語、中国語などのカテゴリの多い言語の文字では、従来の文字認識では、1つの入力文字に対して、数千の辞書内カテゴリとの比較を行わなければならないため、計算量が大きく、処理時間が課題となっていた。

- ・省メモリ化の課題

従来の文字認識では、高精度認識を達成するために大きなメモリが使用できる環境での実行を前提としていた。しかし、近年、スマートフォンなどメモリ量が不十分なデバイスが出現し、高精度認識を実現するためのリソース不足が課題となっていた。また、クラウドなどで並列に多数の認識処理を行う場合にもハードウェア資源が大量に必要なため、個々の認識処理の必要メモリ量を小さくすることは課題である。

このように、近年では、大規模なコンピュータだけではなく、スマートフォンの様な小さな機器で文字認識技術が必要な場合がある。また、大規模なコンピュータでも、大量の画像を処理する必要がある、実用的な文字認識技術は必須の技術である。

このような実用的な文字認識技術で課題となっているのは、認識精度をほとんど低下させないまま処理を高速化できる技術、および、高精度を維持したままの使用メモリ量の削減である。高精度な認識手法をベースとすると、必要なのは高速化、省メモリ化であり、以下の様な解決方法が知られている。

- ・高速化の従来 of 解決方法

高速化 of 課題を解決するために、よく用いられるアプローチとして、2 段階選抜方式がある。すなわち、未知 of 文字が入力されたら、比較的軽い処理を用いて入力文字に類似した辞書内 of カテゴリを選択する大分類を行う。その後、選択されたカテゴリだけを対象に、計算量は大きいが、認識精度が高い手法を用いて詳細に認識を行い、最終的に入力文字に最も類似している辞書カテゴリ 1 つを認識結果として出力する詳細認識を行うものである。

- ・省メモリ化 of 従来 of 解決方法

高精度を維持したまま of 省メモリ化 of 課題を解決するために、本論文では、先ず高精度認識が可能な **MQDF** 詳細認識に注目した。**MQDF** は、各カテゴリ of 分布状況を部分空間として辞書に持ち、それを使った認識を行うため、単純なユークリッド距離値を使用した認識処理よりは高精度な結果が得られる。しかし、メモリに展開される辞書サイズを考慮すると、ユークリッド距離値を使用した認識処理では、各カテゴリ of 平均値情報だけが辞書に必要なのに対し、**MQDF** 認識では、各カテゴリ of 平均値情報だけでなく、各カテゴリ of 分布状況を示す部分空間 of 情報が必要となるため、メモリを非常にたくさん使用していた。

この課題を解決するために **Compact MQDF** 法 [93] [94] では、**MQDF** 辞書の中で大きな割合を占める各カテゴリ of 分布情報を示す特徴ベクトルを小さな部分ベクトルに分割し、それをクラスタリングして類似している部分ベクトルをまとめ、各クラスタにインデックス番号をつける。**MQDF** 辞書 of 分布情報 of ベクトル内容 of 代わりに、近似値として、部分ベクトル単位にインデックス番号で置き換えたものを使用することで、精度を低下させずに辞書サイズを削減している。

しかし、それぞれ of 解決方法でも下 of 様に、まだ課題が残っている。

- ・高速化 of 残存課題

従来 of 2 段階選抜方式 of 大分類処理では、大分類 of 性能についての基準が明確ではなかった。大分類 of 性能を表す基準として、大分類自体 of 処理速度、多くの辞書カテゴリから少数を選択する絞り込み能力、選択した少数 of カテゴリに正解が入っている精度 of 3 つが考えられる。従来 of 大分類処理では、これら of 3 つを同時に満たす手法がないという課題があった。

- ・省メモリ化の残存課題

MQDF 認識処理が高精度を実現できる理由は、特徴空間上の分布が類似している類似文字において、微妙な分布の違いを詳細に記述した分布情報を使用して入力文字と辞書カテゴリ間の評価値を計算するからである。しかし、**Compact MQDF** 法は、無条件で部分ベクトルをクラスタのインデックス番号に置き換えるので、この微妙な分布の違いを同じ物と見なしてしまう可能性が高い。その結果、類似文字間の認識能力が毀損され、オリジナルの **MQDF** 辞書と比較した時に認識精度が低下していた。

このように、実用的な文字認識技術を実現するために、従来技術でも残存課題があった。本論文では、以下の様な改良を行い各課題について解決を図った。

- ・高速化の解決方法

本論文では、この 2 段階選抜方式の大分類処理について、大分類が持つべき 3 つの基準を制定し、この基準を全て満足する新方式として、辞書を構成する学習文字の分布を特徴空間の各軸に反映させたハッシュ手法を提案する。また、本手法を既存の手法と比較して、大分類に関しての有効性を示す。

- ・省メモリ化の解決方法

本論文では、分布情報の部分ベクトルのクラスタリングを **Compact MQDF** 法のように全てのカテゴリに適応するのではなく、類似カテゴリがあるカテゴリには非適応とし **MQDF** の認識精度を保証し、かつ、類似カテゴリが無いカテゴリでは分布情報の部分ベクトルをインデックスで置き換えることで辞書サイズを削減する。このため、辞書サイズを **MQDF** とユークリッド距離の中間のサイズに押さえながら、**MQDF** 認識処理とほぼ同等の認識精度を達成することができる。

1.3 本論文の構成

本論文は、本章を含めて 6 章から成る。

第 1 章では、序論として、本稿の目的について述べた。

第 2 章は、文字認識技術の基礎知識、および、文字認識で使われる主要な概念を提示する。ここでは、先ず、文字認識技術の大まかな処理の流れを示す。具体的には、文字を含む画像のコンピュータへの取得から始まる認識処理過程を説明する。また、この処理の中

で文字認識処理のコア部分である 1 文字認識部分については、複数のカテゴリを含む学習用の文字画像集合から各カテゴリの代表ベクトルを含んだ辞書の作成と、辞書と比較することで入力文字の認識を行う処理について、より詳細に記述する。また、従来の文字認識手法の大まかな分類と、それぞれの分類での概念について簡単に説明する。また、従来の文字認識手法の中で、本論文のベースとなる手法について詳細に記述し、実用的な文字認識にとっての従来技術の課題を再提示する。

第 3 章では、従来技術の課題の 1 つである高精度を維持したままの高速文字認識について解法を与える。従来手法では、高速化手法は、2 段階選抜方式によってなされている。これは、未知の文字が入力されたら、比較的軽い処理を用いて入力文字に類似した辞書内のカテゴリを選択する大分類を行う。その後、選択されたカテゴリだけを対象に、計算量は大きい、認識精度が高い手法を用いて詳細に認識を行い、最終的に入力文字に最も類似している辞書カテゴリ 1 つを認識結果として出力する詳細認識を行うものである。しかし、従来の 2 段階選抜方式の大分類処理では、大分類の性能についての基準が明確ではなかった。大分類の性能を表す基準として、大分類自体の処理速度、多くの辞書カテゴリから少数を選択する絞り込み能力、選択した少数のカテゴリに正解が入っている精度の 3 つが考えられる。従来の大分類処理では、これらの 3 つを同時に満たす手法がないという課題があった。この課題を解決するために、この基準を全て満足する新方式として、辞書を構成する学習文字の分布を特徴空間の各軸に反映させたハッシュ手法を提案する。また、本手法を既存の手法と比較して、大分類に関しての有効性を示す。

第 4 章では、従来技術のもう 1 つの課題である高精度を維持したままの使用メモリの削減方法について解法を与える。ここでは、従来の高精度な認識手法として **MQDF** 法を例にとる。従来の文字認識では、高精度認識を達成するために大きなメモリが使用できる環境での実行を前提としていた。しかし、近年、スマートフォンなどメモリ量が不十分なデバイスが出現し、高精度認識を実現するためのリソース不足が課題となっていた。また、クラウドなどで並列に多数の認識処理を行う場合にもハードウェア資源が大量に必要なため、個々の認識処理の必要メモリ量を小さくすることは課題である。この課題を解決するために **Compact MQDF** 法では、**MQDF** 辞書の中で大きな割合を占める各カテゴリの分布情報を示す固有ベクトルを小さな部分ベクトルに分割し、それをクラスタリングして類似している部分ベクトルをまとめ、各クラスにインデックス番号をつける。**MQDF** 辞書の分布情報のベクトル内容の代わりに、近似値として、部分ベクトル単位にインデックス番号で置き換えたものを使用することで、精度を低下させずに辞書サイズを削減して

いる。**MQDF** 認識処理が高精度を実現できる理由は、特徴空間上の分布が類似している類似文字において、微妙な分布の違いを詳細に記述した分布情報を使用して入力文字と辞書カテゴリ間の評価値を計算するからである。しかし、**Compact MQDF** 法は、無条件で部分ベクトルをクラスターのインデックス番号に置き換えるので、この微妙な分布の違いを同じ物と見なしてしまう可能性が高い。その結果、類似文字間の認識能力が毀損され、オリジナルの **MQDF** 辞書と比較した時に認識精度が低下するという課題があった。

本章では、分布情報の部分ベクトルのクラスタリングを **Compact MQDF** 法のように全てのカテゴリに適応するのではなく、類似カテゴリがあるカテゴリには非適応とし **MQDF** の認識精度を保証し、かつ、類似カテゴリが無いカテゴリでは分布情報の部分ベクトルをインデックスで置き換えることで辞書サイズを削減する。これにより、辞書サイズを **MQDF** とユークリッド距離の中間のサイズに押さえながら、**MQDF** 認識処理とほぼ同等の認識精度を達成することができることを示す。

第 5 章では、4 章までで開発した技術を映像中のテロップ認識へ適用した検証実験について記述する。また、特にテロップの中で、1 行の中で文字の色が変化している場合についても読み取れるように、テロップ文字画像を色分解し、各色レイヤー上で文字領域を求め、最後に色レイヤーをまたいだ **DP** による最適な文字列パターン抽出と認識についての評価結果を提示する。

第 6 章は、本論文の結論で、本研究で得られた成果を総括し、本研究が取り組んだ課題に関する研究の将来展望について検討する。

第 2 章

文字認識技術の基礎知識、主要な概念

2.1 文字認識技術の基礎知識

デジタルコンピュータを使用した光学的文字認識の基本プロセスは、図 2.1 のようなものである。

1) 画像入力

スキャナまたはデジタルカメラによって文字列を含んだカラー画像が撮影され、コンピュータに入力される

2) 前処理

2-1) 2 値化

カラー画像を 2 値画像に変換する（文字部分が黒、それ以外が白になるようにする）

2-2) 雑音除去

以下の処理を行うのに障害となる文字列以外の黒画素連結領域を削除する

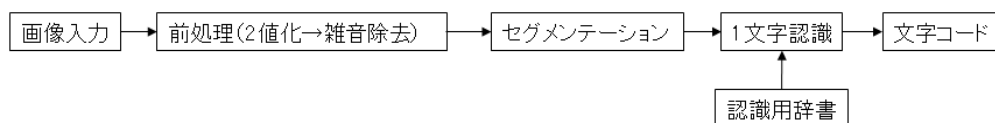


図 2.1 文字認識処理フロー



図 2.2 1 文字認識処理フロー

3) セグメンテーション

画像の中の各文字の領域を求める

4) 1 文字認識

各文字の領域内部を認識用辞書と比較して 1 文字ずつ文字認識して、文字コードを獲得し、それを並べて認識結果のテキスト文字を出力する。

この中で、文字認識のコア部分は、4 の処理である。この処理の中で、辞書を使用するため、先に、辞書の作成処理について説明する（図 2.2）。

辞書は、辞書内に持つ各文字カテゴリ毎に、多数の学習文字を予め用意しておき、そのカテゴリ毎の文字集合から各カテゴリの代表情報などの辞書情報を求め、カテゴリの文字コードと共に保存することで作成される。

4-1) 辞書作成処理

この処理は、以下の処理から構成される。

a) 正規化

1 つのカテゴリの文字画像でもいろいろな形、大きさがある。それを揃えて、文字画像の変形を吸収するために画像変形を行う。これを正規化と呼んでいる。

b) 特徴抽出

2 値の文字画像から特定の特徴量の数値を抽出し、文字を数値列（特徴ベクトル）で表現する。

c) 代表ベクトル、付随情報作成

1 つのカテゴリで多数の異なる文字画像を収集し、個々の文字画像から特徴ベクトルを抽出する。多数の特徴ベクトルから代表ベクトルを作成する。同時に高精度化のための付随情報（例えば、部分空間情報等）を抽出する

d) 辞書作成

カテゴリを表す文字コードと、代表ベクトル、付随情報を 1 セットとして保存する。す

すべてのカテゴリについて同様の情報を保存して、辞書を作成する

次に、未知の文字画像が入力された時の文字認識処理は、以下のような処理である。

4-2) 文字認識処理

a) 正規化

入力された文字画像を、学習した辞書の文字画像に近づけるために、辞書作成時と同じ正規化処理を行う。

b) 特徴抽出

辞書作成時と同じ特徴抽出処理を行い、特徴ベクトルを抽出する。

c) マッチング

入力文字画像から求めた特徴ベクトルと、辞書内の代表ベクトルの1つとの間の類似度または距離を求め、その代表ベクトルの文字コードとともに記憶する。これを辞書内の全てのカテゴリの代表ベクトルについて行い、類似度（距離値）と文字コードのペアの集合を作成する。

類似度（距離値）と文字コードのペアの集合を対象に、類似度または距離値の大きさに並び替え（類似度の場合は大きい順に、距離値の場合は小さい順に並び替える）、順位が第1位のものを入力文字に最も類似したものであると判定し、そのペアの中の文字コードを認識結果として出力する

この1文字認識処理手法は、大きく分けて2種類の手法に分類できる。1つは、構造解析手法で、もう片方は重ね合わせ手法と呼ばれる。各手法は、特徴抽出方法と、抽出した特徴量に依存したマッチング手法によって分類される。それぞれの手法の特徴は以下の通りである。

1) 構造解析手法

概要：文字画像からストロークなどの文字を構成する構造部分特徴を特徴量として抽出し、それらの位置関係などを使ってマッチングを行うアプローチ

特徴量：文字ストローク、文字内のループ、ストロークの端点（エンドポイント）、ストローク交差点などの種類とその位置

マッチング手法：入力文字から求めた上記特徴量と辞書内の特徴量の一致度を使用してマッチングする。特徴量の相対的な位置関係の類似度も使用される。

2) 重ね合わせ手法 (マッチング手法)

概要：文字画像を小さな領域に分割し、小領域の中の特徴を統計的に表現、数値化した特徴量を求める。それを並べた特徴ベクトルを特徴量とする。マッチングは、辞書内のカテゴリの代表ベクトルと入力文字の特徴ベクトルの類似性を類似度または距離値で評価して行うアプローチ

特徴としては、画像の画素値そのものを使う画素特徴、文字画像のストロークの中心線や、文字の輪郭の方向を使う方向特徴などが知られている。

また、マッチング手法は、下に記述するように、多数の手法が知られている。

この2種類の分類の中で、構造解析法は、雑音に弱く、安定したストロークを抽出できないという課題があり、重ね合わせ手法が主流となった。本論文では、以降、重ね合わせ手法に基づいて論じていく。重ね合わせ手法の特徴量で、輪郭の方向特徴を使ったものの例として、「加重方向ヒストグラム法」とよばれる方法について説明する (図 2.3)。

a) 文字画像を固定サイズ (64 × 64 画素) に正規化後、2 値化して文字が黒、背景が白画素とする

b) 文字画像の輪郭に注目し、輪郭画素の各画素に4方向のコード (|、一、／、＼) を与える

c) 画像を分割して、各小ブロックに入る輪郭画素の数を4つの方向毎に集計し、この小ブロック内の4次元のベクトルを作成する

d) 各ブロックの部分特徴ベクトルを対象に、ガウシアンフィルタを用いて、近接するブロックをまとめて、ブロック内の部分特徴ベクトルの同じ方向の成分をぼかす処理を行い、近接ブロックをまとめた新たな部分ブロックの特徴ベクトルを作成する。これは1つの代表ベクトルで変形した文字画像も対応するためである。

e) 特徴ベクトル作成

新たに作った部分ブロックの順番に4次元ベクトルを並べて、全体で64次元のベクトルを作成する

2.2 高精度文字認識の基礎知識

高精度な文字認識を行う上で、文字認識処理の中で、最も重要なのがマッチング部分である。マッチング部分は、大きく2つに分類できる。1つは、文字画像を特徴ベクトルに変換した後、同様に変換された辞書の中の各カテゴリ代表の特徴ベクトルと、ベクトル間

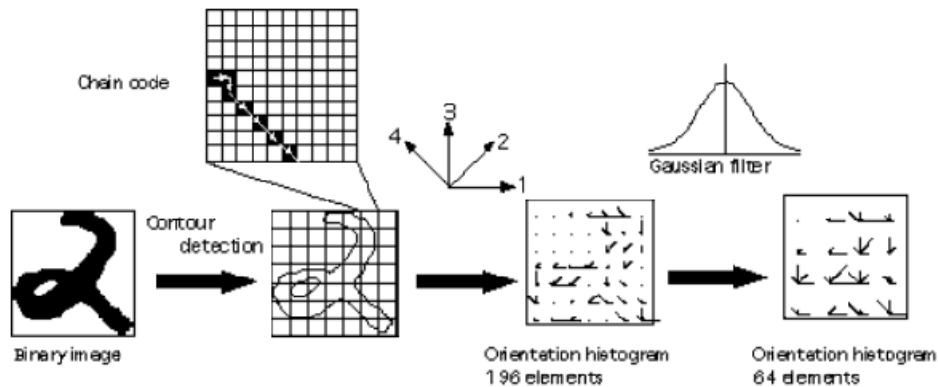


図 2.3 加重方向ヒストグラム特徴作成フロー

の類似度、または距離を求めて、最も類似しているカテゴリを認識結果とするもの。もう 1 つは、各カテゴリの特徴空間上の分布を求めておき、2 つのカテゴリまたは複数のカテゴリ間の識別境界を辞書に記述しておき、入力文字の特徴ベクトルの特徴空間上の位置と、カテゴリ間の識別境界により認識するものである。これまでの研究で、様々な高精度認識手法が考案されてきた。代表的な手法を以下に示す。

2.2.1 類似度、距離値を使う手法

この手法は、辞書内のカテゴリの代表ベクトルと、入力文字の特徴ベクトルとの間の類似度、または、距離を以下のそれぞれの計算式で求め、最も小さい距離値の、または、最も大きな類似度の辞書内カテゴリを認識結果として求める手法である。

この手法の中で、比較的高精度となるものは、マハラノビス距離法、MQDF 法、複合類似度法、部分空間法である。

シティブロック距離

入力文字の特徴ベクトルとカテゴリ代表ベクトルの対応する要素の差分の絶対値の総和で表される距離値。

これは、特徴空間内で、入力文字の特徴ベクトルとカテゴリ代表ベクトルで表される 2 点間を、空間を規定する軸に平行な線に沿って動かした場合の距離値。

$$D_{cityblock} = \sum_{i=1}^d |x_i - \mu_i^C| \quad (2.1)$$

ここで、

x_i : 入力文字の d 次元特徴ベクトルの第 i 要素

μ_i^C : 辞書内のカテゴリ C の d 次元代表ベクトルの第 i 要素

ユークリッド距離

特徴空間内で、入力文字の特徴ベクトルとカテゴリ代表ベクトルで表される2点間の直線距離値。

$$D_{euclid} = \sqrt{\sum_{i=1}^d (x_i - \mu_i^C)^2} \quad (2.2)$$

ここで、

x_i : 入力文字の d 次元特徴ベクトルの第 i 要素

μ_i^C : 辞書内のカテゴリ C の d 次元代表ベクトルの第 i 要素

マハラノビス距離

カテゴリの分布形状を考慮したときのカテゴリ代表点と入力文字ベクトルの間の距離値。

$$D_{maharanobis} = \sum_{i=1}^d \frac{1}{\lambda_i} (\phi_i \cdot (x - \mu^C))^2 \quad (2.3)$$

ここで、

x : 入力文字の d 次元特徴ベクトル

μ^C : 辞書内のカテゴリ C の d 次元代表ベクトル

λ_i : カテゴリ C の共分散行列の第 i 番目の固有値 ($i = 1, 2, \dots, d$)

ϕ_i : カテゴリ C の共分散行列の第 i 番目の d 次元固有ベクトル ($i = 1, 2, \dots, d$)

MQDF

ベイズ則の識別関数で、カテゴリの確率密度関数を正規分布と仮定し、対数をとって表される QDF において、固有値の大きい方から $k+1$ 番目以降で固有値を固定値としたもの。

$$D_{MQDF} = \frac{1}{h^2} \left[\|x - \mu^C\|^2 - \sum_{i=1}^k \left(1 - \frac{h^2}{\lambda_i}\right) (\phi_i \cdot (x - \mu^C))^2 \right] \quad (2.4)$$

$$+ \sum_{i=1}^k \log \lambda_i + (d - k) \log h^2$$

ここで、

x : 入力文字の d 次元特徴ベクトル

μ^C : 辞書内のカテゴリ C の d 次元代表ベクトル

λ_i : カテゴリ C の共分散行列の第 i 番目の固有値 ($i = 1, 2, \dots, d$)

ϕ_i : カテゴリ C の共分散行列の第 i 番目の d 次元固有ベクトル ($i = 1, 2, \dots, d$)

k : 実験的に求められる切断次元数 ($k \leq d$)

h^2 : 実験的に求められる定数

単純類似度

これは、入力文字の特徴ベクトルと辞書内のカテゴリの代表ベクトルとの作る角度を評価するもの。

$$S_s = \frac{(x \cdot \mu^C)^2}{\|x\|^2 \|\mu^C\|^2} \quad (2.5)$$

ここで、

x : 入力文字の d 次元特徴ベクトル

μ^C : 辞書内のカテゴリ C の d 次元代表ベクトル

複合類似度

これは、各カテゴリの分布を示す直行軸を考え、入力ベクトルをその軸へ投影した距離値の2乗に、その軸の固有値を重みとしてかけた値の全軸の総和の値を正規化したものである。

$$S_c = \sum_{i=1}^d \frac{\lambda_i (\phi_i \cdot x)^2}{\lambda_1 \|x\|^2} \quad (2.6)$$

ここで、

x : 入力文字の d 次元特徴ベクトル

λ_i : カテゴリ C の共分散行列の第 i 番目の固有値 ($i = 1, 2, \dots, d$)

ϕ_i : カテゴリ C の共分散行列の第 i 番目の d 次元固有ベクトル ($i = 1, 2, \dots, d$)

部分空間法

これは、各カテゴリの分布を示す直行軸を考え、入力ベクトルをその軸へ投影した距離値の2乗を一定軸数まで総和した値である。

$$S_p = \sum_{i=1}^k (\phi_i \cdot x)^2 \quad (2.7)$$

ここで、

x : 入力文字の d 次元特徴ベクトル

ϕ_i : カテゴリ C の共分散行列の第 i 番目の d 次元固有ベクトル ($i = 1, 2, \dots, d$)

k : 実験的に求められる切断次元数 ($k \leq d$)

2.2.2 カテゴリ間の識別境界を使う手法

この手法は、各カテゴリの特徴空間上の分布を求めて、2つのカテゴリまたは複数のカテゴリ間の識別境界を辞書に記述しておく。認識処理では、入力文字の特徴ベクトルの特徴空間上の位置と、カテゴリ間の識別境界の位置関係または、判別結果により認識するものである。これらの手法は、すべて比較的高精度の認識結果を得ることができる。

ニューラルネットワーク

生物の神経細胞を模した情報処理システムで、文字画像の画素値を入力する入力層、中間層、認識結果を表示する出力層の3層のネットワーク構成がよく使われる (図 2.4)。入力層に学習用の文字を入力し、出力層において入力した文字のカテゴリのノードにだけ1で、その他が0となるように、層間を結ぶ線の重み係数を変更する。これを学習するすべての文字について行って、重み係数を決定する。この3層間のノードを結ぶ線の重み係数が辞書に相当する。

認識処理は、学習済みのニューラルネットワークの入力層に未知入力文字を入力して、出力層の最も大きく反応したノードのカテゴリを認識結果とする。

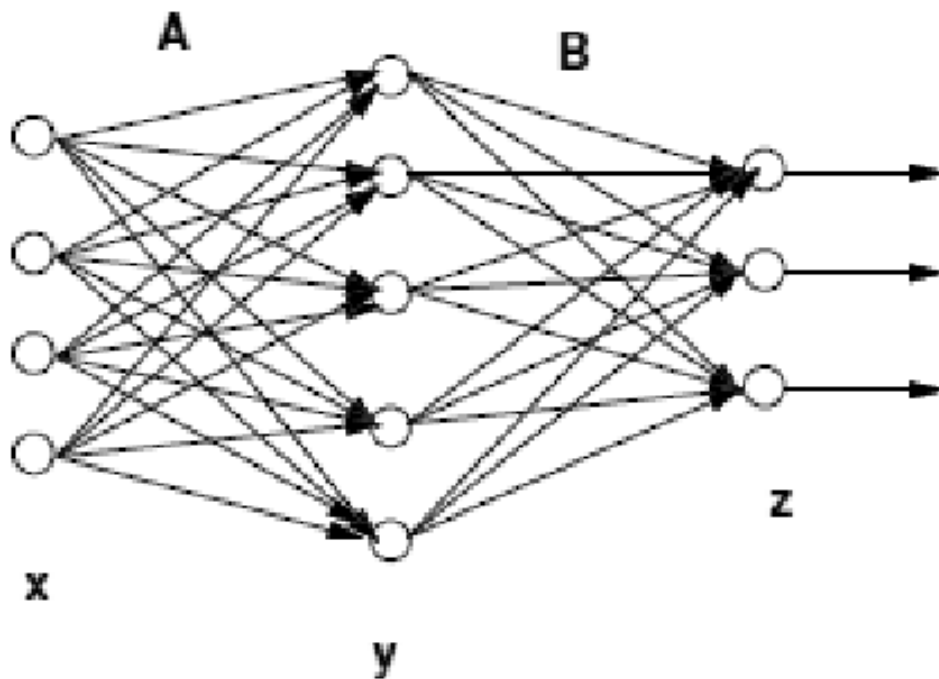


図 2.4 ニューラルネットワーク

Support Vector Machine (SVM)

サポートベクターマシンは、2つのカテゴリの学習サンプルを用い、2カテゴリを識別する超平面を求めて、これを使用して2カテゴリの識別を行う認識手法である（図 2.5）。識別面は、学習サンプルの各データ点との距離が最大となるマージン最大化超平面を求めて決定する。

ここでマージンとは、超平面から学習サンプルの各点への距離の最小値を言い、このマージンを最大にしながら学習サンプルの各点を二つのカテゴリで分類しようとする、結局片方のカテゴリに属する点との距離の中の最小値と、もう片方のカテゴリに属する点との距離の中の最小値とが等しくなるように超平面が位置しなければならず、このような超平面をマージン最大の超平面という。線形超平面で分離できない場合は、カーネル関数を用いてパターンを有限もしくは無限次元の別の特徴空間へ写像し、その特徴空間上で線形分離を行うことにより対応する。SVMでは、マージン最大化により、未学習データに対して高い識別性能を得ることができる。

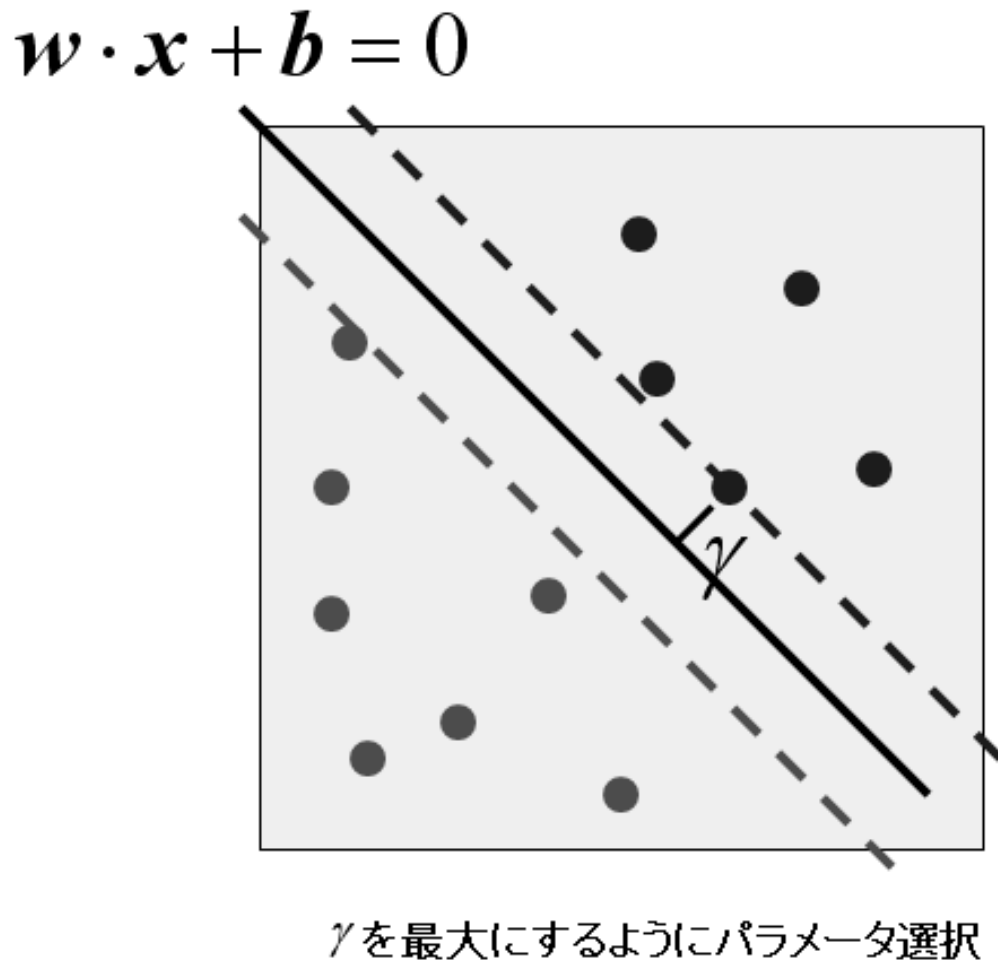


図 2.5 SVM

Boosting

ブースティングは、教師あり学習を実行するための機械学習のメタアルゴリズムである。すなわち、一般の（任意の）機械学習アルゴリズムをもっと賢く実行するためのフレームワークと言える。弱い学習機をまとめて、重みを付けてその結果を足し合わせることで、強い学習機を生成することを目指している。

学習サンプルに重み付けをして、それを対象に、1種類の機械学習アルゴリズムで学習し、1種類の分類器を作る。誤分類されるサンプルは重みを増し、正しく分類される例は重みを減らし、学習サンプルに対する重み付けを変更して、もう1回同じ学習アルゴリズムで学習し、別の分類器を作成する。この処理を繰り返すことで、複数の分類器が作成さ

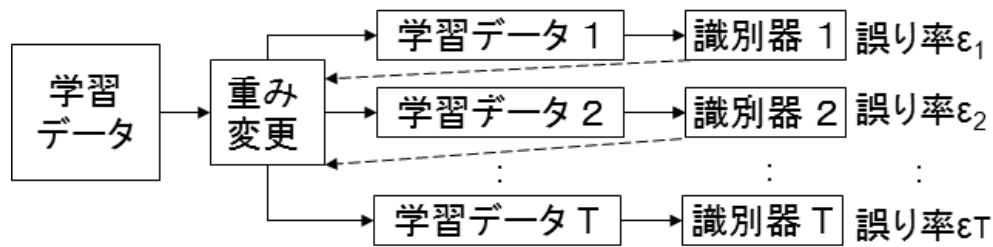


図 2.6 Ada Boost の学習過程

れる。これにより 1 つの分類器では認識できない特異な学習サンプルも他の分類器で認識できるので、全体としては、特異点を包含することができる非線形な識別面を構成できる。

ブースティングの中の 1 つの AdaBoost のアルゴリズムは、以下のようなものである (図 2.6)。

1. サンプル (特徴ベクトルと教師信号であるカテゴリの組) への重みは、最初は $1/N$ (N はサンプル数) で初期化する
2. T 個の弱識別器を作る
3. (まずは、1 つ目の) 弱識別器 t を作る。弱識別器 t の作り方は、重み付きのサンプルを利用して、一般的な機械学習アルゴリズムを利用して学習する
4. 弱識別器 t の誤り率を求める。誤り率は、各サンプルを見たときに、サンプルのカテゴリと弱識別器の出力カテゴリが一致しないサンプルの重みを足し合わせたものになる
5. 誤り率から弱識別器 t の信頼度を求める。具体的には、誤り率が小さいほど、信頼度が大きくなるようにする
6. サンプルへの重みを更新する。サンプルへの重みは、弱識別器 t が正しく識別できたサンプルは重みが低くなるよう、弱識別器 t が間違って識別したサンプルは重みが高くなるように更新する
7. サンプルの重みの和が 1 になるように正規化する
8. 3. に戻って、 $t + 1$ 個目の弱識別器を作る。 T 個作れたらアルゴリズム終了。
9. 最終的な識別は、全ての弱識別器を信頼度で重みづけして多数決を取る (図 2.7)

2.3 従来の高精度認識手法の課題

以上が、従来の高精度文字認識手法の概要であるが、これまでの高精度文字認識技術の研究では、ハードウェアの制約をあまり考慮しないで、主に高精度化を目指した研究が

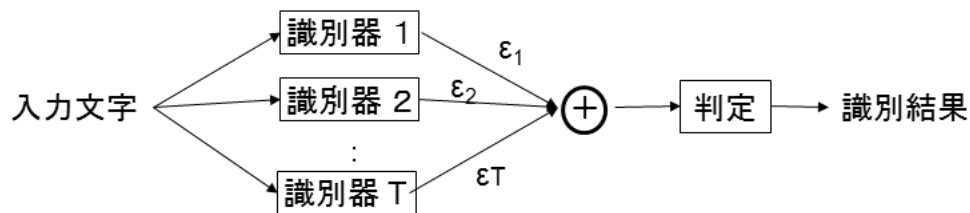


図 2.7 Ada Boost の識別過程

主流であった。例えば、学習サンプル文字の分布を記述した辞書を使用した部分空間法、MQDF 法は、入力ベクトルとカテゴリの代表ベクトルとの距離値（類似度）を単純に求めるだけでなく、学習サンプルの分布を考慮した精密な計算を行うため、複雑な演算が必要である。そのため、高速処理が難しい課題がある。また、これらの手法は、学習サンプル文字の分布を記述するために辞書サイズが大きくなる課題がある。一方、近年では、大規模なコンピュータだけではなく、スマートフォンの様な小さく処理能力が低い機器で文字認識技術が必要な場合がある。また、大規模なコンピュータでも、大量の画像を処理するために並列に多数の処理を行い、ハードウェア資源が大量に必要な場合がある。このため、高精度文字認識の高速化、省メモリ化は必須の技術である。

この様に、従来は、実用的な文字認識技術の研究では不十分であった。ここで、実用的な文字認識技術とは、高精度を維持したままの高速処理、かつ、コンピュータのメモリサイズの制限がある状況での高精度認識処理である。

このような実用的な文字認識技術で課題となっているのは、認識精度をほとんど低下させないまま処理を高速化できる技術、および、高精度を維持したままの使用メモリ量の削減である。従来の文字認識手法の実用性の観点からの課題は、以下のようにまとめられる。

・高速化の課題

日本語、中国語などのカテゴリの多い言語の文字では、従来の文字認識では、1つの入力文字パターンに対して、数千の辞書内カテゴリとの比較を行わなければならないため、計算量が大きく、処理時間が課題となっていた。このような場合の認識処理の高速化が求められていた。

・省メモリ化の課題

従来の文字認識では、高精度認識を達成するために大きなメモリが使用できる環境での実行を前提としていた。しかし、近年、スマートフォンなどメモリ量が不十分なデバイス

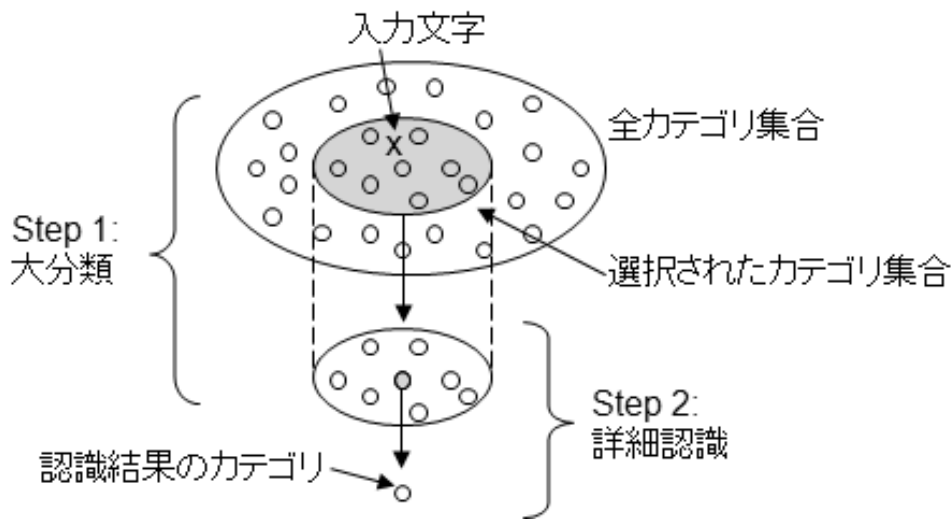


図 2.8 2段階選抜方式

が出現し、高精度認識を実現するためのリソース不足が課題となっていた。

以下、2つの課題を解決するための従来技術の基礎知識を示す。

2.4 高速化の基礎知識

高速化の課題を解決するために、よく用いられるアプローチとして、2段階選抜方式 [95] がある。すなわち、未知の文字が入力されたら、比較的軽い処理を用いて入力文字に類似した辞書内のカテゴリを選択する大分類を行う。その後、選択されたカテゴリだけを対象に、計算量は大きいが高認識精度が高い手法を用いて詳細に認識を行い、最終的に入力文字に最も類似している辞書カテゴリ 1 つを認識結果として出力する詳細認識を行うものである (図 2.8)。

2.5 省メモリ化の基礎知識

上記の認識手法の中で、高精度認識が可能なものは、シティブロック距離値やユークリッド距離値を使用した単純な手法以外のものである。それらの高精度認識手法では、高精度を達成するために、類似度、距離値を使う手法では、特徴空間内で1つのカテゴリの学習サンプル集合の分布形状情報を使用している。また、カテゴリ間の識別境界を使う手法では、各種の重みを使用している。このように、高精度認識手法では、複雑な情報を辞

書として持つ必要があり、辞書サイズは大きくなる。

シティブロック距離値法やユークリッド距離値法では、各カテゴリの代表ベクトルだけで済んでいたため辞書サイズは、非常に小さいものであった。しかし、例えば MQDF 法では、各カテゴリの代表ベクトルだけでなく、各カテゴリの特徴空間内での分布形状を示す、固有値、固有ベクトルの情報が必要となる。代表ベクトルの次元を d 次元とすると、固有値は k ($\leq d$) 次元ベクトル、固有ベクトルは、 d 次元ベクトルが k 個のベクトル列となる。 d 、 k の値が大きくなると、MQDF 法での辞書サイズは、ユークリッド距離値法などの辞書サイズと比べて大幅に大きくなる事がわかる。

この課題を解決するために、Long らの Compact MQDF 手法 [93] がある。これは、以下の様な手法である。

1) MQDF 認識手法をベースとして、各カテゴリの分布情報を示す固有ベクトルを部分ベクトルに分割し、全ての部分ベクトルを対象としたクラスタリングを行い、クラスタのインデックス番号と各クラスタの代表ベクトルのテーブルを作成しておく。

2) オリジナルの MQDF 辞書の代わりに固有ベクトルの部分ベクトルをクラスタのインデックス番号に置き換えて辞書サイズを削減する。

3) MQDF 計算の時は、クラスタのインデックス番号を代表ベクトルに置き換えることで擬似的な固有ベクトルを作成し、それを使用して評価値の計算を行う

MQDF 等の大きなサイズの辞書は、この手法のように固有ベクトルを近似値に置き換えることで、サイズを小さくすることができる。

2.6 従来技術の課題と本論文のアプローチ

・高速化について

2段階選抜では、第1段階目の大分類が非常に重要になる。大分類は、辞書カテゴリから少数のカテゴリを選択する時に、大分類が持つべき3つの指標が考えられる。従来の大分類技術では、この3つの指標を同時に満足する物が無かった。

本論文では、この3指標を全て、同時に満足する新方式として、辞書を構成する学習文字の分布を特徴空間の各軸に反映させたハッシュ手法を提案する。また、本手法を既存の手法と比較して、大分類に関しての有効性を示す。

・省メモリ化について

MQDF 手法において、省メモリ化のための Long らの手法では、固有ベクトルに近似値を使用することで辞書サイズを削減していた。MQDF 認識処理が高精度を実現できる理

由は、特徴空間上の分布が類似している類似文字において、微妙な分布の違いを詳細に記述した分布情報である固有値、固有ベクトルを使用して入力文字と辞書カテゴリ間の評価値を計算するからである。しかし、Long らの手法は、無条件で部分ベクトルをクラスタのインデックス番号に置き換えるので、この微妙な分布の違いを同じ物と見なしてしまう可能性が高い。その結果、類似文字間の認識能力が毀損され、オリジナルの MQDF 辞書と比較した時に認識精度が低下していた。

本論文では、分布情報の部分ベクトルのクラスタリングを全てのカテゴリに適応するのではなく、類似カテゴリがあるカテゴリには非適応とし MQDF の認識精度を保証し、かつ、類似カテゴリが無いカテゴリでは分布情報の固有ベクトルをインデックスで置き換えることで辞書サイズを削減する。このため、辞書サイズを MQDF とユークリッド距離の中間のサイズに押さえながら、MQDF 認識処理とほぼ同等の認識精度を達成することができる。

第 3 章

高速化のための研究成果

3.1 はじめに

近年、文書スキャナの処理スピードが非常に向上してきた。そのため、短い時間で大量の文書が処理できるようになってきた。会社では、文書を、エンタープライズ・コンテンツ管理ソフトウェアのような文書管理システム（DMS）へ入力するためにそのような高速スキャナを使用している。この場合、新しい文書を入力する時の処理の高速性だけが重要ではなく、例えば会計監査の場合のように、データを早く使用するために入力した文書を効率的に検索する能力も重要である。それ故、高精度で、かつ、高速な OCR システムがそのような検索処理のために熱望されている。日本語を対象とした OCR では、高速な OCR システムの設計は特に難しい。なぜならば、日本語には、およそ 5000 カテゴリの文字があるからである。マッチング処理の処理時間を低減することは、効率的な日本語 OCR を開発する上で、最も重要な課題である。高速 OCR 処理を達成するために、高精度を維持したまま処理時間を短縮できる 2 段階選抜方式 [95] が効率的である。この手法では、図 3.1 に示す様に、認識処理は、2 つのステージの処理から構成される。すなわち、ステップ 1 では、大分類処理が、辞書中の全カテゴリ集合の中から、入力文字に類似している少数のカテゴリだけを候補カテゴリとして選択する。次に、ステップ 2 として詳細認識処理で、選択された少数の候補カテゴリ集合の中から、1 つの最終認識結果カテゴリを選択、出力する。

大分類処理は、正しい文字カテゴリを選択したカテゴリ集合に高い確率で入れながら、かつ、入力文字に類似したカテゴリだけになるように、詳細認識の対象となる選択カテゴリ数を劇的に小さくすることができる。このように、大分類は、全体の処理時間の低減

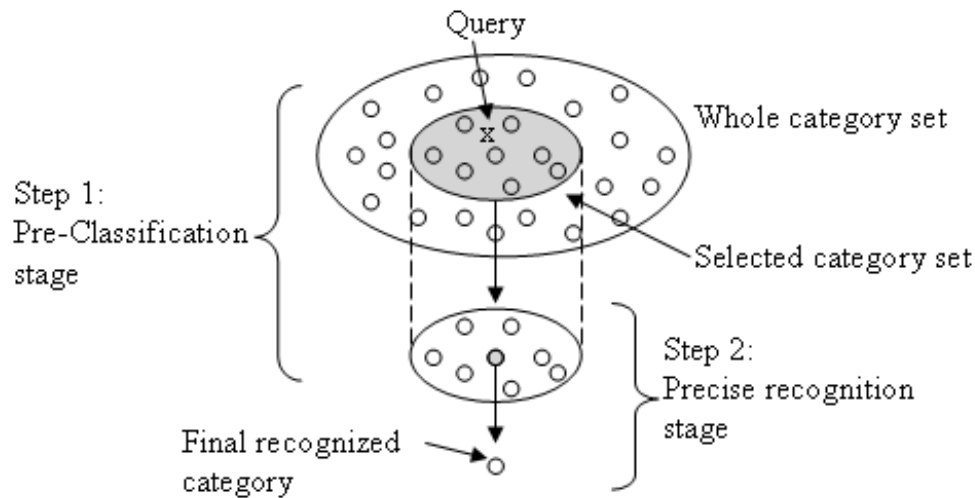


図 3.1 2段階選抜方式

と、同時に認識精度についても非常に大きな役割を担っている。

日本語の **OCR** を考える場合、大分類が本来持つべき3つの本質的な要求項目が考えられる。これは、以下のようなものである。

1) 非常に高い分類正解率

大分類は、詳細認識の前処理であるので、選択したカテゴリ集合にほとんど **100%** の確率で正しい文字が入る必要がある。

2) 大分類自体の高速性

認識処理の全体の処理時間は、大分類の処理時間と詳細認識の処理時間の合計である。全体の処理時間を短くするためには、大分類自身の処理時間が短い必要がある。

3) 選択したカテゴリ数の小ささ

通常、詳細認識処理は、複雑な計算を行うため、1つのマッチング処理時間が長い傾向にある。大分類で選択した候補カテゴリ集合の中のカテゴリ数が大きい場合、詳細認識の処理時間が大幅に大きくなり、全体の認識処理が遅くなってしまう。処理全体を高速化するために、大分類で選択するカテゴリ数は小さいことが求められる。

この3つの要求項目の中で、分類正解率と選択したカテゴリ数の小ささの間にはトレードオフの関係がある。

これまでに、いくつかの手法がこの問題を解決する目的で考案されてきた。

Nearest Neighbor (NN) 探索手法は、特徴空間内で、入力文字に近い上位 k 個の文字を候補として選択することから、大分類としてみなすことができる。他の大分類として使用できる従来手法としては、**VQ クラスタリング**、**ANN**、**LSH**、**Associative Matching (AM)** 法などが提案されている。通常の **k-NN** [96] 手法では、入力文字から特徴ベクトルを求め、クエリベクトルと辞書内の各カテゴリの代表ベクトル間の距離が計算される。その後、上位 k 個の辞書内のカテゴリが大分類結果として選択される。**K-means** や **LVQ** をベースとしたクラスタリング手法 [97] では、クラスタリングは認識処理とは別にオフラインで行われ、図 3.2 に示す様に特徴空間でお互いに類似しているカテゴリが同じクラスタに所属するようにクラスタが作成される。距離計算は、クラスタの代表ベクトルとクエリベクトルの間で行われるため、距離計算の回数を削減することができる。図 3.2 で、カテゴリ A、B、C の学習サンプルベクトルは、図のようにまとめられる。入力クエリベクトル X に最も近いクラスタ中心を持つクラスタが選択され、そのクラスタ内にある複数のカテゴリが大分類の結果として選択される。この手法は、通常の **k-NN** 手法よりも高速であるが、クエリベクトルと各クラスタの代表ベクトルの間の距離計算が必要なため、後に記述するハッシュ法に比べれば遅い。

クラスタリングを行う他の手法として、木構造をベースとした、**kd-tree** [96] [98] [99] [100] [101]、**R-tree** [99] [100] [102]、**SR-tree** [99] [103]、**Sp-tree** [104]、**MPA** [105]、**ANN** [99] [100] [101] [106] [107] などがある。この内、**ANN** 手法は、学習時に特徴空間を分割しながら 2 分木の木構造 **kd-tree** を作り、**kd-tree** 構造により特徴空間を多数のセルに分割して管理する辞書を使用している。特徴空間内で、入力クエリベクトルがプロットされた点から一定距離以内にあるセルのカテゴリが大分類結果として選択される（図 3.3）。図 3.3 で、入力クエリベクトル X は、カテゴリ C のセルにプロットされている。この場合、 X とセル内の学習ベクトル C との間の距離が計算され、クエリベクトル X を中心とし、この距離値を半径とする超球が設定される。この超球と重なるセルのカテゴリが大分類の結果として選択される。上で定義した大分類が持つべき 3 つの要求項目については、分割セルによって特徴空間を完全にカバーするため、**ANN** 手法は、非常に高い分類正解率を達成できる。しかし、この手法では、**kd-tree** 構造の中で入力クエリベクトルがプロットされるセルを求めることや、プロットされた位置から近隣のセルを求めるために、多数の識別計算が必要であり、処理時間はハッシュ法と比べて大幅に遅くなる。

Locality-Sensitive Hashing (LSH) 手法 [100] [101] [108] [109] [110] [111] [112] [113]

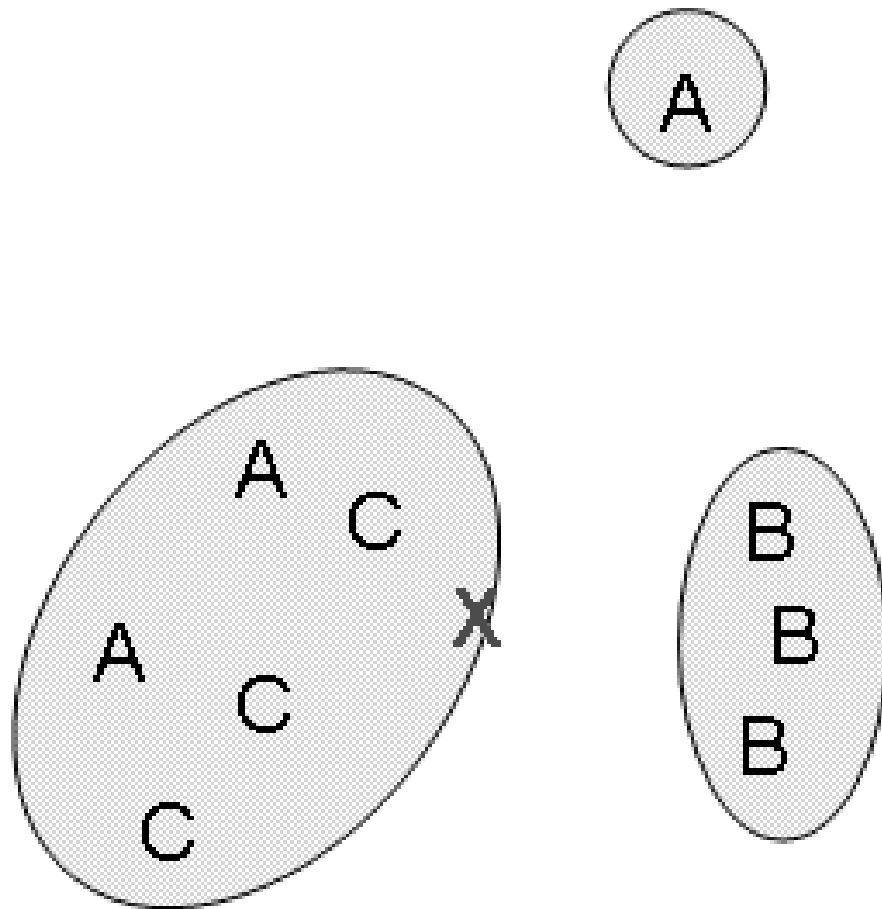


図 3.2 VQ クラスタリング手法の例

は、特徴ベクトルを、ハッシュ関数を使ってランダムに生成された複数の低次元の空間に投影する。ハッシュテーブルは、この低次元空間上に構築される。クエリベクトルが入力されると、ハッシュ関数が入力ベクトルを複数のハッシュテーブルに投影し、投影されたハッシュテーブルに登録されているカテゴリを統合して大分類結果とする。このとき、複数のハッシュテーブルの中で、少なくとも1つのハッシュテーブルに登録されているカテゴリは、大分類結果として出力される。図 3.4 は、2次元のランダムに選択された部分空間の軸と、その上に構築された **bucket** と呼ばれるハッシュテーブルを示す。各軸の上でクエリベクトル X が投影された **bucket** が網掛けで表現されている。この網掛け部分の **bucket** に入るカテゴリが大分類の結果として選択される。大分類が持つべき3つの要求項目については、この手法は、候補カテゴリの選択のための距離計算を行わず、代わりに投影処理と論理 OR 演算だけを使用するので、上の2つの手法より高速であることがわか

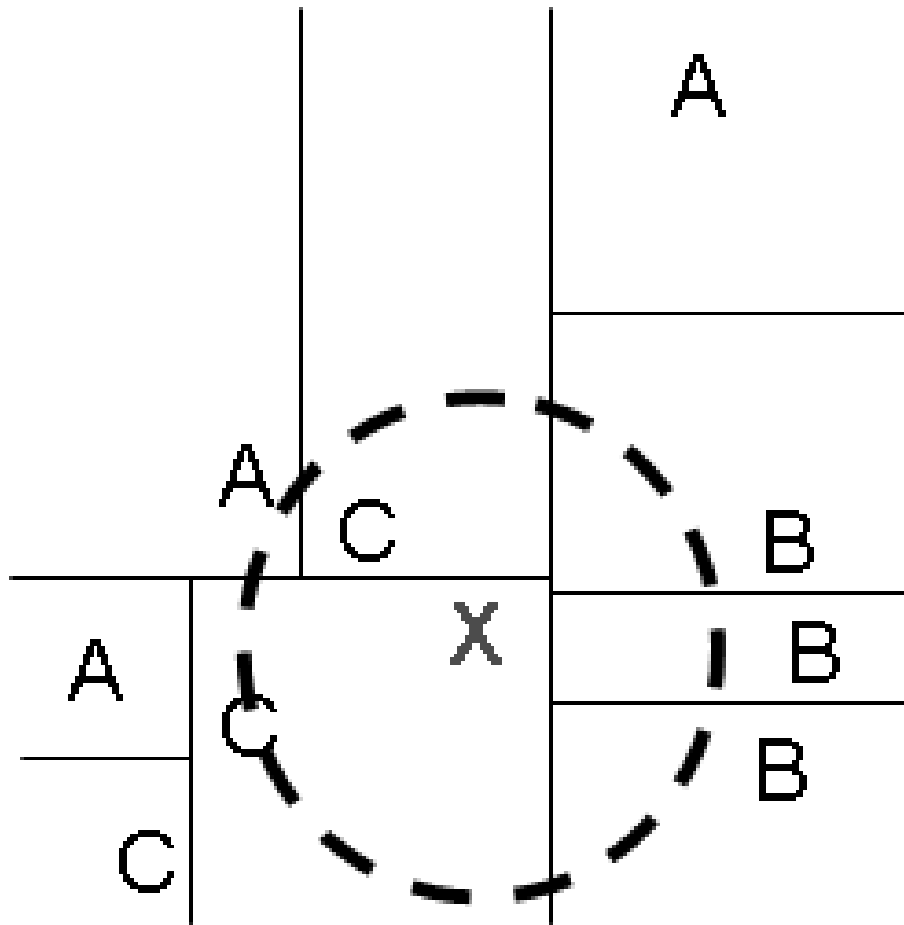


図 3.3 ANN 手法の例

る。また、この手法では、多数のランダムに選択された直行でない軸を使用するために、非常に高い分類正解率を達成することが可能である。しかし、このランダムで直行でない軸の使用と、論理OR演算の使用により、選択されるカテゴリの数は、他の手法より大きくなってしまう。

Associative Matching (AM) 手法 [114] は、直交軸の上に作られた、ビット配列で表現されたハッシュテーブルを使用する手法である。軸上での各カテゴリの範囲は、そのカテゴリの学習サンプルベクトルを軸上に投影した時の最大値と最小値で決められた範囲を、一定のマージンだけ外側に拡張した範囲である。各軸上の範囲を総合すると、1つのカテゴリは、特徴空間上にある区間を形成する。大分類処理の時には、入力クエリベクトルが投影された場合に、特徴空間上の区間が投影点を包含するカテゴリを全て抽出し、これを大分類結果として出力する（図 3.5）。図 3.5 では、特徴空間上でのカテゴリ A、B、C の

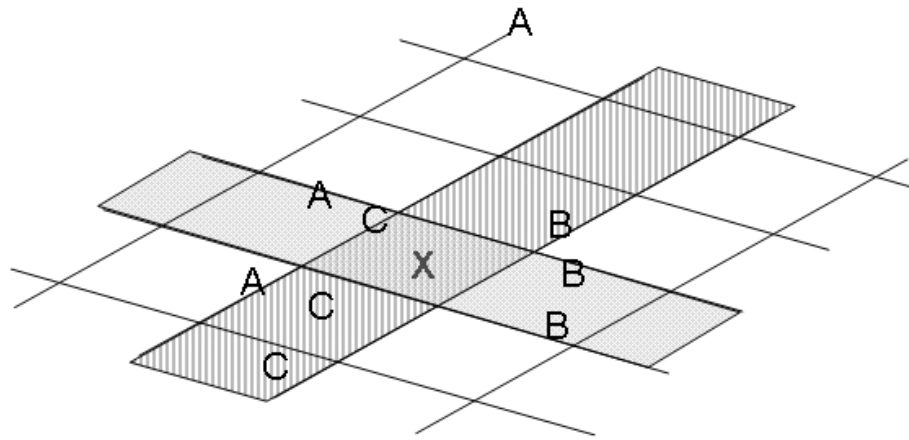


図 3.4 LSH 手法の例

表 3.1 従来手法の3要件の評価比較

Methods	VQ Clustering	ANN	LSH	AM
分類正解率	△	○	○	○
大分類処理時間	△	×	○	○
選択カテゴリ数の小ささ	△	○	×	△

範囲を表す3つの矩形がある。各矩形の範囲は、2つの軸へ各カテゴリの学習サンプルを投影した時の最大値、最小値によって求まる範囲に一定のマージンを加えたものとして定義される。入力クエリベクトルが投影された点Xを包含する矩形のカテゴリが大分類結果として選択される。大分類が持つべき3つの要求項目に関しては、この手法は、高い分類精度と、上に記述した他の手法より高速な処理が可能である。これは、各軸のハッシュテーブルの辞書がビット配列で表現されており、かつ、大分類処理のために、距離計算や識別処理より高速な論理 AND 演算だけしか使用されないからである。しかし、この手法では、多くのカテゴリを候補として選択してしまう。なぜなら、各軸上の範囲は最大値と最小値だけから決定されるので、学習サンプルの特徴空間上の実際の分布を反映した範囲より大きくなってしまからである。

従来手法をまとめると、ハッシュ法は、VQ クラスタリング手法や ANN 手法よりも処理速度に関して有利である。これは、ハッシュ法は、より単純な演算を使用するからである。さらに、AM 手法は、直交軸を使用し、ハッシュテーブルは各カテゴリの分布を考慮

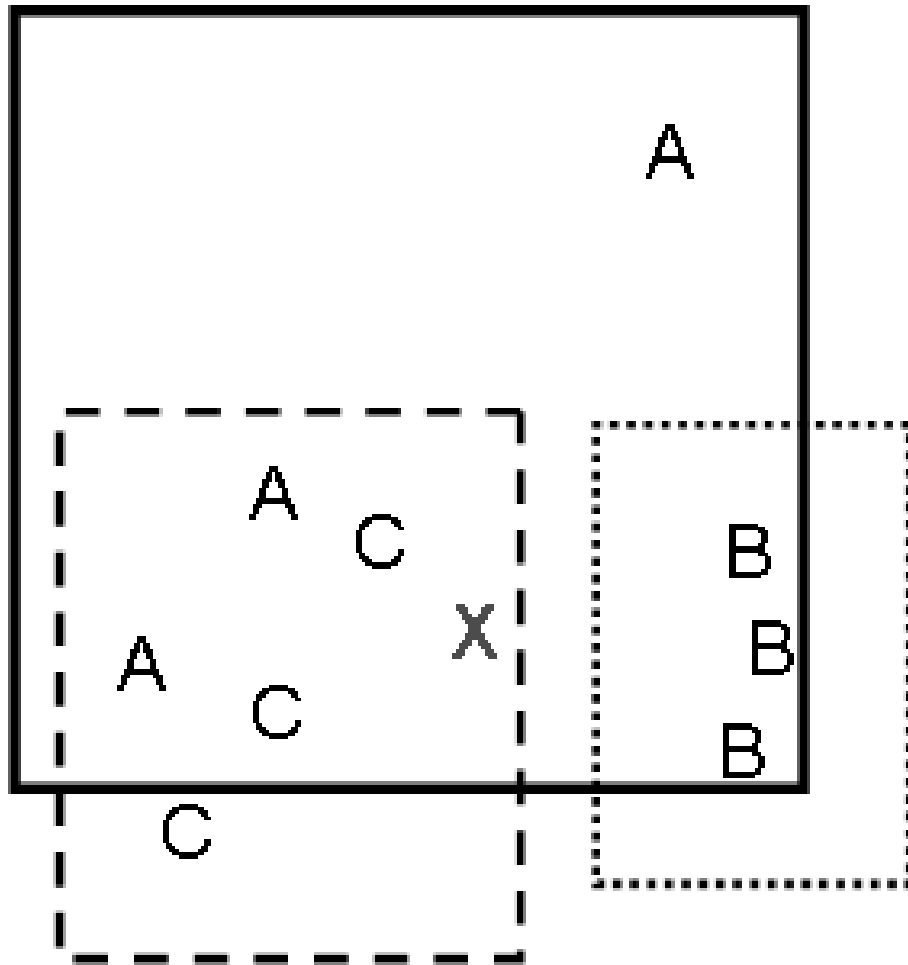


図 3.5 AM 手法の例

して設計されるため、この手法は **LSH** 手法より優れている。各要求項目について各手法を 3 段階で評価した結果を表 3.1 に示す。この表ではどの手法も全ての項目で○となるものがないことがわかる。すなわち、従来手法は、大分類が持つべき 3 つの要求項目を同時に満足しない。この課題を解決するために、本論文では、**Margin Added Hashing** と名付けた新しい大分類手法を提案する。この手法の特徴は、特徴空間の中で各カテゴリの範囲を縮小することができることと、ビット演算による高速処理が可能であるということである。この **Margin Added Hashing** のアイデアは、以前発行された文献 [115] に記述があるが、本論文では、詳細なパラメータを用いた **LSH**、**ANN** との新しい比較評価と詳細な考察を行った。

3.2 Margin Added Hashing (MAH)

従来の大分類手法と、大分類が持つべき3つの要求項を考えると、Associative Matching (AM) 手法が、高い分類性能とビット演算による処理時間の短さにおいて、最も優れていると判断できる。しかし、AM 法は、大分類結果として多くのカテゴリを選択してしまうため、認識処理全体の処理時間が比較的大きくなるという欠点があった。特徴空間内の各カテゴリの実際の範囲を考慮することで、AM 法をベースとして大分類正解率を保ったまま全体の認識処理時間を縮小する新しい大分類手法 Margin Added Hashing (MAH) 手法を提案する。MAH 手法では、大分類のために、AM 法と同様に特徴空間の各軸上のハッシュテーブルを使用する。AM 法での各軸上の最大値、最小値で決まる範囲と異なり、MAH 法での各ハッシュテーブルは、各カテゴリの学習サンプルベクトルを軸へ投影した点にマージンを加える手法で定義される。ここで、マージンは、入力文字画像の変動を吸収するために使用されるものである。この方法により、各軸上の各カテゴリの範囲は AM 法のものよりも小さくすることができるので、MAH 法では、正しいカテゴリを AM 法より少ない数の選択候補カテゴリの中に入れることができる。さらに、MAH 法は、AM 法と同様に直交軸とハッシュテーブルのビット表現を用いているために、大分類の処理を非常に短い時間で行うことができる。図 3.6 は、1つの軸のうえのハッシュテーブルの例である。本論文では、マージン値は定数を使用している。図 3.6 で、 $bit1_{\omega k}$ 配列は、式 3.1, 3.2, 3.3 と表 3.2 で定義され、ハッシュテーブル辞書の第 k 軸上のカテゴリ ω のビット配列を表している。

$$bit1_{\omega k}^j = (\{b_{\omega k}^j\}_1, \{b_{\omega k}^j\}_2, \dots, \{b_{\omega k}^j\}_{max_index}) \in B^{max_index} \quad (3.1)$$

$$\{b_{\omega k}^j\}_i = \begin{cases} 1 & \text{for } |(x_{\omega}^j \cdot \phi_k) - i| \leq margin \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

$$bit1_{\omega k} = bit1_{\omega k}^1 \vee bit1_{\omega k}^2 \vee \dots \vee bit1_{\omega k}^{N_{\omega}} \quad (3.3)$$

ここで、

$$\begin{aligned} B &= \{0, 1\} \\ \mathbf{u} &= (u_1, u_2, \dots, u_n) \in B^n, \\ \mathbf{v} &= (v_1, v_2, \dots, v_n) \in B^n \end{aligned} \quad (3.4)$$

表 3.2 記号の意味

表記	意味
k	軸のインデックス
ϕ_k	第 k 番目の軸の固有ベクトル
ω	カテゴリ, $\omega = 1, 2, \dots, \text{max_category}$.
x_ω^j	カテゴリ ω の第 j 番目の学習サンプルの特徴ベクトル
N_ω	カテゴリ ω の学習サンプルの特徴ベクトル数
i	1つの軸のビット配列へのインデックス番号, $i = 1, 2, \dots, \text{max_index}$.
max_index	全ての軸で同じ値である最大インデックス番号
max_axis	特徴ベクトルの次元数

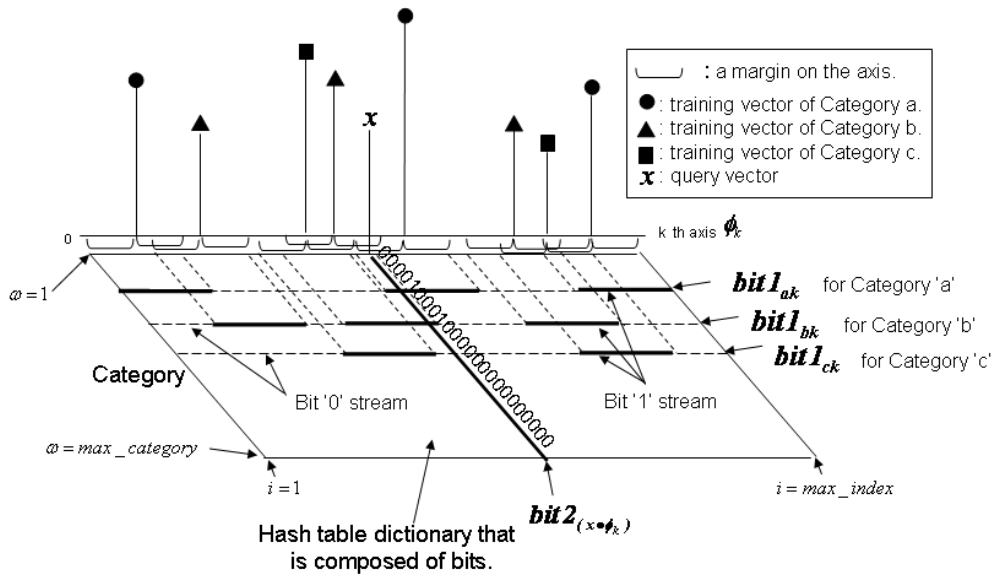


図 3.6 MAH 手法の辞書

$$\begin{aligned}
 \mathbf{u} \vee \mathbf{v} &= (u_1 \vee v_1, u_2 \vee v_2, \dots, u_n \vee v_n) \\
 \mathbf{u} \wedge \mathbf{v} &= (u_1 \wedge v_1, u_2 \wedge v_2, \dots, u_n \wedge v_n)
 \end{aligned}
 \tag{3.5}$$

であり、式 3.4 の、要素が 0 または 1 である論理ベクトル \mathbf{u} と \mathbf{v} について、式 3.5 に示す様な 2 つの演算を定義する。即ち、演算 ' \vee ' は、2 つの論理ベクトル間のビット OR 演算を表す。この演算では、2 つのベクトルの対応する要素間のビット OR 演算が行われる。演算 ' \wedge ' は、2 つの論理ベクトル間のビット AND 演算を表し、同様に、この演算で

は、2つのベクトルの対応する要素間のビット AND 演算が行われる。本論文では、論理ベクトルをビット配列としてみなしている。

$\mathbf{bit1}_{\omega k}^j$ は、カテゴリ ω の第 j 番目の学習サンプルベクトル x_{ω}^j の第 k 軸でのビット配列を表している。カテゴリ ω の学習サンプル x_{ω}^j が入力された場合、これは第 k 軸の上の $\lfloor (x_{\omega}^j \cdot \phi_k) \rfloor$ の値に投影される。次に、ビット配列 $\mathbf{bit1}_{\omega k}^j$ の $\lfloor (x_{\omega}^j \cdot \phi_k) \rfloor - \text{margin}$ から $\lfloor (x_{\omega}^j \cdot \phi_k) \rfloor + \text{margin}$ の範囲の間の全てのビットが '1' に設定される。この範囲の外側のビットは全て '0' に設定される。カテゴリ ω の各学習サンプルベクトルの各ビット配列は、論理 OR 演算でマージされ、第 k 軸のカテゴリ ω のビット配列 $\mathbf{bit1}_{\omega k}$ がハッシュテーブル辞書で作成される。例えば、カテゴリ a, b, c の各ビット配列 $\mathbf{bit1}$ は、図 3.6 の様に表現される。ここで、ビット配列の中の太い線はビット '1' が連続した部分を表し、点線は、ビット '0' が連続した部分を表現している。ビット '1' が連続した部分は、各カテゴリ毎の各学習サンプルベクトルによって設定されるので、MAH 法は、各カテゴリについて軸上の最大値、最小値で決まる範囲を使用している AM 法よりもカテゴリの存在範囲を狭くすることが可能である。

MAH 法の大分類処理は、AM 法と同様な手法で行われる。すなわち、クエリベクトル x が入力されると、第 k 軸の $\lfloor (x \cdot \phi_k) \rfloor$ の値に投影される。第 k 軸上で、この値を持つカテゴリの集合は、各ビットが 1つのカテゴリに対応するビット配列 $\mathbf{bit2}_{\lfloor (x \cdot \phi_k) \rfloor}$ で表される（図 3.6 を参照）。ここで、 $\{b_{\lfloor (x \cdot \phi_k) \rfloor}\}_i$ は、ビット配列 $\mathbf{bit2}_{\lfloor (x \cdot \phi_k) \rfloor}$ の上で、入力クエリベクトルが投影された第 k 軸上の点 $\lfloor (x \cdot \phi_k) \rfloor$ の第 i 番目のカテゴリを表すビットである。MAH 法の最終的な大分類結果は、式 3.6, 3.7 に示す様に、各軸の $\mathbf{bit2}_{\lfloor (x \cdot \phi_k) \rfloor}$ 配列の論理 AND 演算の結果として求められる。即ち、*Output_bit* 配列の中のビット '1' で示されるカテゴリが最終的な MAH 法の大分類結果出力である。

$$\mathbf{bit2}_{\lfloor (x \cdot \phi_k) \rfloor} = (\{b_{\lfloor (x \cdot \phi_k) \rfloor}\}_1, \{b_{\lfloor (x \cdot \phi_k) \rfloor}\}_2, \dots, \{b_{\lfloor (x \cdot \phi_k) \rfloor}\}_{\text{max_category}}) \in B^{\text{max_category}} \quad (3.6)$$

$$\mathbf{Output_bit} = \mathbf{bit2}_{\lfloor (x \cdot \phi_1) \rfloor} \wedge \mathbf{bit2}_{\lfloor (x \cdot \phi_2) \rfloor} \wedge \dots \wedge \mathbf{bit2}_{\lfloor (x \cdot \phi_{\text{max_axis}}) \rfloor} \quad (3.7)$$

MAH 法を使用することで、大分類は、特徴空間上で、入力クエリベクトルがプロットされる範囲のカテゴリだけをとりえることができるようになる。従って、MAH 法は、AM 法よりも少ない数の候補カテゴリを選択しながら、AM 法と同じ処理スピードで、同程度の分類精度を維持することができる。本手法の 2 次元での辞書の表現は、図 3.7 のよ

うになる。図 3.7 では、カテゴリ A、B、C の各カテゴリの存在範囲を示す矩形が描かれている。まず、各カテゴリの学習サンプルを各軸に投影して、それにマージンを加えることで、軸上のカテゴリの存在範囲が求められる。次に、特徴空間内の最終的なカテゴリの範囲が各軸の範囲の交差部分の矩形として求められる。文字が入力され、クエリベクトルがプロットされた場合、矩形がこのプロット点を包含するカテゴリが、大分類結果のカテゴリとして選択される。図 3.5 の AM 法と比較すると、本手法では各カテゴリのカバーする範囲は AM 法のものよりも小さく、かつ、学習サンプルの存在範囲を全てカバーしていることがわかる。本手法では、正解のカテゴリが C である入力ベクトル X は、カテゴリ C の矩形内だけにプロットされているので、正しいカテゴリ C だけが分類結果として選択される。しかし、図 3.5 の AM 法では、同じ位置にプロットされる入力ベクトルは、カテゴリ A と C の矩形の中にはいるので、カテゴリ C だけでなく A も候補として選択されてしまうことになる。図 3.5 と図 3.7 の違いは 2 次元では小さいが、実際に使用される高次元空間を考慮すると、その差は大きくなる。この場合、提案手法は、AM 法と比較してほとんど同じ分類精度を維持しながら、AM 法よりも少ない候補カテゴリを選択することが可能となる。

3.3 評価実験

提案する MAH 手法の有効性を確認するために、日本語印刷文字画像を使った評価実験を行った。文字画像は、先ず正規化され、次に文字画像の輪郭が抽出される。輪郭の画素は、水平方向からの 0, 45, 90, 135 度の 4 方向に分類される。次に文字画像がいくつかのセルに分割され、各セル内の方向毎の画素数を計測して並べることによって、最初の特徴ベクトルが作成される [116]。この特徴ベクトルを元に直交軸へ投影し、16 次元に次元圧縮した特徴ベクトルを作成し、これを大分類用の特徴ベクトルとして使用した。この特徴ベクトルの 1 つの要素は 1 バイトで表現されるので、1 つの要素のダイナミックレンジは 1 から 256 である。しかし、ダイナミックレンジを小さくするために、要素の値を、固定値によって割って使用している。本論文では、MAH 法と AM 法に使用される固定値として 2 を採用した。これによって、各軸上の取り得る範囲は、1 から 128 となるので、表 3.1 の *max_index* は、128 となる。

学習用の文字画像は、22 から 124 種類のフォントを使用して作成されたものである。1 つのカテゴリにおいて、各フォントは 1 回だけ使用したので、1 つのカテゴリの学習には同数の文字画像が使用された。全カテゴリを総合した学習文字数は、445518 である。

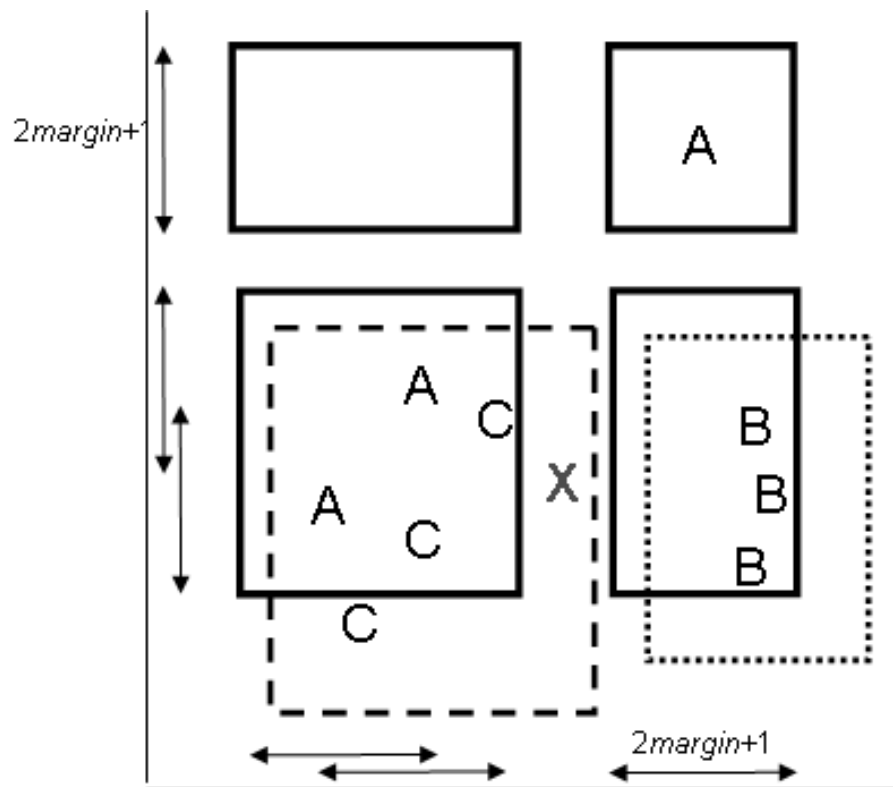


図 3.7 MAH 手法の例

いくつかのカテゴリは、特徴空間上で複雑な分布形状をしているので、非漢字文字について最大3つに分割するようにクラスタリング手法を導入した。本論文では、クラスタリング手法として、事前の実験により効率が良く、高い認識精度が得られる理由から、[117]の手法を使用した。この結果、全学習サンプル文字から作られるクラスタの数は、5225となり、本論文では、本来のカテゴリの代わりとして、このクラスタをカテゴリとしてみなすこととした。従って、日本語印刷文字の 5225 カテゴリが最終的に使用される。

このクラスタリングにおいて、非漢字だけを対象として、3以下のクラスタに分割したのは、以下のような理由による。すなわち、非漢字文字のストロークの自由度は、漢字文字のそれよりも高いので、非漢字文字の変形は、漢字文字の変形よりも大きい。従って、非漢字文字の特徴空間内での分布は、漢字文字の分布よりも複雑になる。実際、辞書として、全ての文字が1つの代表ベクトルだけを使用する場合、非漢字文字の認識精度は、漢字文字の精度より悪くなり、非漢字文字だけにクラスタリングを行い、分布を分割することで、文字認識率が向上した。このように、特徴空間での各カテゴリの分布のレベルを揃えるために、クラスタリングを非漢字文字だけに導入した。

評価実験は、以下の評価項目について行なった。

1) 大分類の分類エラー率

大分類で選択したカテゴリ集合の中に正しい入力文字のカテゴリが入っている場合に、入力文字は正しく分類されたと見なされる。大分類の分類エラー率 ER は、以下の式によって求められる。

$$ER = 1 - \frac{N_{cs}}{N_{ci}} \quad (3.8)$$

ここで、 N_{cs} は、正しく分類された文字数、 N_{ci} は、入力文字数を表す。

2) 大分類と全体の認識処理の処理時間

ここでは、大分類だけの処理時間と、大分類と詳細認識の処理時間を合計した全体の認識処理の処理時間を計測する。この評価では、詳細認識として Cityblock 距離を使用する単純な文字認識処理を使用した。処理時間は、Core2Duo E8500 の CPU, 2GB のメモリを持つ Windows XP のパソコンで計測した。

3) カテゴリ選択率

1つの文字が入力された場合、いくつかの候補カテゴリが大分類処理によって選択される。全ての評価用文字を入力した場合の出力カテゴリ数の平均値を計算し、この平均値を辞書内の総カテゴリ数 5225 で割った数値をカテゴリ選択率 SR として求め、これを評価項目として使用する。 SR は、以下の式で定義される。

$$SR = \frac{N_{as}}{N_{cd}} \quad (3.9)$$

ここで、 N_{as} は、選択出力されたカテゴリ数の平均値であり、 N_{cd} は辞書の中の全カテゴリ数で 5225 である。

評価実験は、MAH 法だけでなく、比較評価として AM 法、VQ クラスタリング手法、ANN 法、LSH 法でも行った。本論文では、一般的なクラスタリング手法であり、クラスタリング処理の処理時間が短いことから、VQ クラスタリング手法として k-means 法 [118] を使用した。初期のシードベクトルとしては、事前実験によって決められた 20, 60, 100 個のランダムなベクトルを使用した。候補カテゴリの選択処理では、入力クエリベクトルに近い上位 N_c 個のクラスタを選択し、その中に入っているカテゴリ全てを大分類選択結

果とした。事前実験によって、 N_c は5とした。MAH手法とAM手法では、マージンの値がパラメータになるが、マージン値を振らせて、よい性能を提示するマージン値を探した。ANN法では、ライブラリ [107] で提供される **k-nearest neighbor** 探索を使用した。この評価では、パラメータとして、選択ベクトルの数 k と近似ファクタを表す eps 値を変化させて評価を行った。kd-tree 構造を構築するための学習サンプルの中には、同じカテゴリの文字が多数あるので、ANN の出力の選択カテゴリ集合のなかのユニークなカテゴリの種類は、非常に小さくなる。大分類後の詳細認識では、ANN の出力のカテゴリ集合からユニークなカテゴリだけを取り出すためのソート処理が必要になる。このソート処理時間も ANN の処理時間の中に含んで計測した。LSH法では、本論文では E2LSH(Exact Euclidean LSH) [99] [100] [101] [110] [113] を使用した。この手法のパラメータは、 L 、 k 、 w であり、それぞれ最もよい性能を探すために変化させて調査した。ここで、 L は、ハッシュテーブルの数、 k は部分空間へ圧縮するときの部分空間の次元数、 w は、各軸でのハッシュサイズを表す。

学習サンプルの文字とは別に、116528 文字の日本語印刷文字画像を評価実験用に使用した。これらの文字画像は、70 種類の実際の文書をスキャンした画像から、手動で切り出して収集したものである。詳細認識の処理時間は、大分類結果の候補文字数に比例するため、詳細認識の処理時間は、1 文字の詳細認識の処理時間に、大分類の結果のカテゴリ数をかけることによって計算で求めた。事前の Cityblock 距離値を使用した単純な詳細認識処理の実験によって、1 文字の詳細認識の処理時間は、0.067msec. / 文字とした。

大分類の評価結果を図 3.8 に示す。各手法は、それぞれのパラメータがあり、大分類として最もよいバランスとなる場合を探すために、各パラメータを変動させた結果を図 3.8 は示している。MAH手法とAM手法では、パラメータはマージン値であり、これは、図 3.8 では m として表記されている。VQ クラスタリング手法では、パラメータは、クラスタ数 n である。ANN手法では、パラメータは eps と k である。いずれもライブラリ [107] で用いられるもので、 eps は、近似ファクタを、 k は、**k-nearest neighbor** 法で探される最近傍ベクトルの数を表す。LSH法では、パラメータは、 L 、 k 、 w である。 L は、ハッシュテーブルの数、 k は部分空間へ圧縮するときの部分空間の次元数、 w は、各軸でのハッシュサイズを表す。図 3.8 は、msec. で表される認識の全処理時間と、大分類の分類エラー率の関係を表している。この図で、横軸は処理時間を、縦軸は分類エラー率を示す。図 3.8 でわかるように、よい手法は、分類エラー率が小さく、かつ同時に、処理時間が短いものである。この図で、認識の全処理時間と大分類の分類エラー率の間で、最もよ

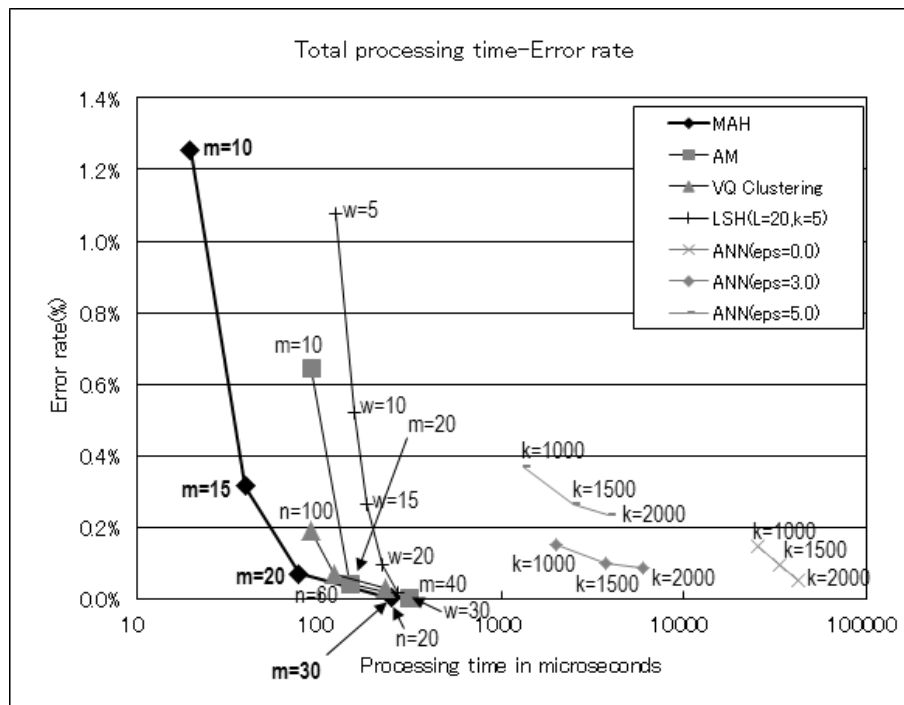


図 3.8 認識処理全体の処理時間と分類エラー率の関係

いバランスをとることができる手法は、MAH 法であることがわかる。

3.4 考察

図 3.8 の結果を分析するために、分類エラー率とカテゴリ選択率の関係、および、分類エラー率と大分類処理時間の関係の評価した。結果を図 3.9、および図 3.10 に示す。図 3.9 で、横軸はカテゴリ選択率を、縦軸は分類エラー率を示す。図 3.10 で、横軸は大分類だけの処理時間を、縦軸は分類エラー率を示す。図 3.8 において、全認識処理時間と分類エラー率の間にはトレードオフの関係があることがわかる。同様に、図 3.9 から、カテゴリ選択率と分類エラー率の間にもトレードオフの関係があることが分かる。図 3.9 において、ANN 法は、低い分類エラー率のままカテゴリ選択率を小さくできる観点において、今回評価した手法の中で最も優れていることがわかる。ANN 法は、特徴空間全体を区切って隙間なく kd-tree 構造の辞書を作成するので、学習サンプルにないような例外的な文字画像が入力された場合には、その文字が kd-tree 構造の大きなセルの中にプロットされることになる。入力クエリベクトルとそのセルの学習ベクトルとの間の距離が大きくなることにより、ANN 法は、プロットされた点を中心に広い区間にある学習サンプルベクトル

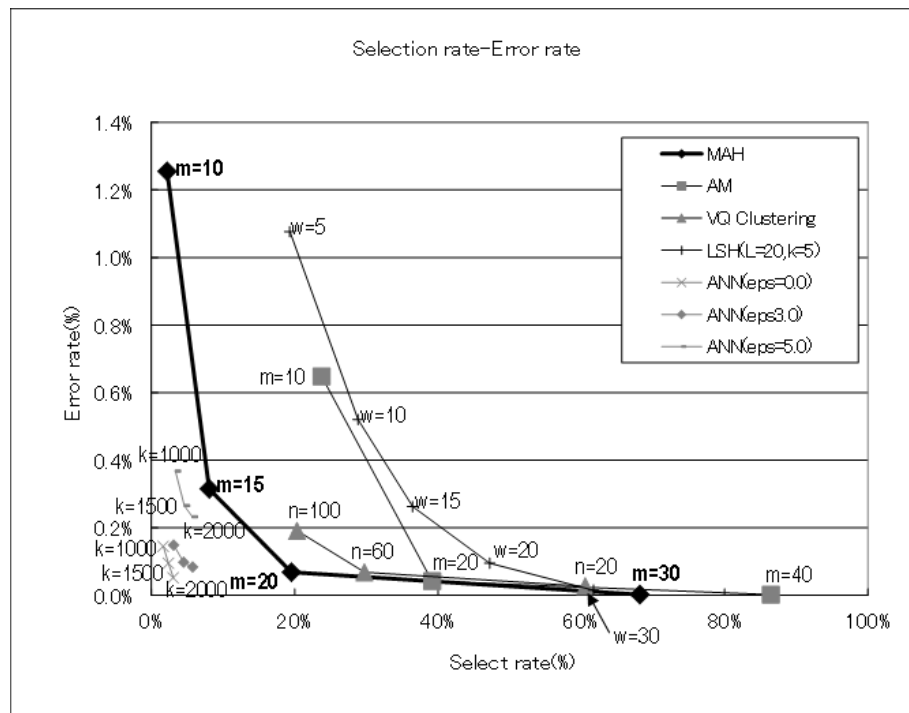


図 3.9 カテゴリ選択率と分類エラー率の関係

のカテゴリを収集するので、そこに正しいカテゴリが入る確率が高まるためであると思われる。LSH 法は、カテゴリ選択率において、評価手法の中で最も悪い性能となった。これは、LSH 法では、ハッシュテーブルとしてランダムな軸を多数使用し、各軸で収集したカテゴリの論理和を最終的な選択結果とするために、大分類結果として選択されるカテゴリ数が大きくなる傾向にあるからである。

図 3.9 では、ANN 法が最もよい手法であったが、大分類だけの処理時間の項目を見ると、図 3.10 にみられるように ANN 法は最も悪い手法となる。他の大分類手法と比較して、ANN 法は、およそ 100 倍以上遅い。kd-tree 辞書は、たくさんのカテゴリの学習サンプルから作られるので、kd-tree 辞書は多くのレイヤーを持つようになる。ANN 法では、ルートノードから末端のセルノードまで、入力ベクトルが近い方を識別しながら木構造を辿る処理が必要なので、他の手法より複雑な計算が必要となる。さらに、ANN 法では、通常の処理では、選択カテゴリ集合の中のユニークカテゴリは極めて少ない。そこで、ANN 法は、選択するカテゴリ集合にユニークカテゴリが多くなり、正しいカテゴリがその中に入るように、入力クエリベクトルの近く的大量の学習ベクトルを選択しなければならない。これらの理由により、ANN 法の大分類だけの処理時間は、他の手法に比べて大変大

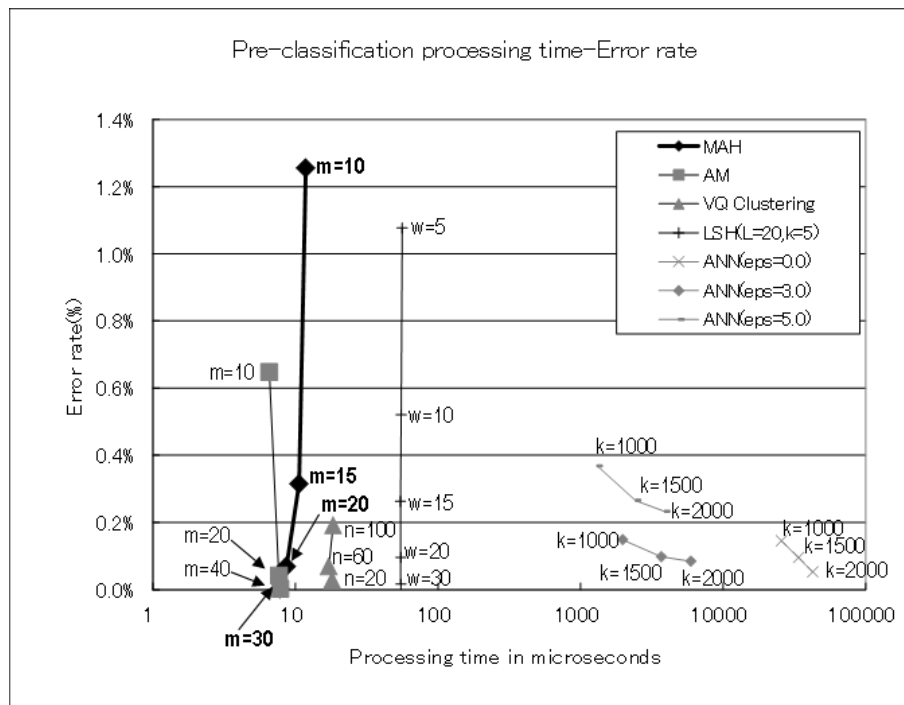


図 3.10 大分類処理だけの処理時間と分類エラー率の関係

きなものになってしまう。図 3.10 で示されるように、LSH の大分類だけの処理時間もあまり小さくない。これは、LSH 法ではランダムな軸をハッシュテーブルとして使用しているため、冗長な処理となっているからである。

図 3.9 に示されるカテゴリ選択率に関して、MAH 法は、AM 法や VQ クラスタリング法よりもよい結果となっている。VQ クラスタリング法については、1 つのクラスタに入るカテゴリの数と分類エラー率との間にトレードオフの関係があるので、高い大分類精度を維持したまま選択カテゴリ数を小さくすることは難しい。さらに、AM 法において、特徴空間でのカテゴリの分布が図 3.5 に示されるように広がっている場合は、特徴空間でのこのカテゴリの範囲は非常に大きなものになってしまう、大分類で選択したカテゴリ集合に正しくないカテゴリが多数入り込む可能性が高くなる。しかし、図 3.10 では、これらの 3 手法の大分類だけの処理時間は、ANN 法や LSH 法より小さいことが分かる。これは、VQ クラスタリング法は、ANN 法の計算より単純なユークリッド距離の計算だけを使用し、AM 法と MAH 法は、より簡単なビット演算だけを使用しているからである。さらに、AM 法と MAH 法は、LSH 法で使われるランダムな軸より効率のよい直交軸を使用していることから処理時間が短くなる。

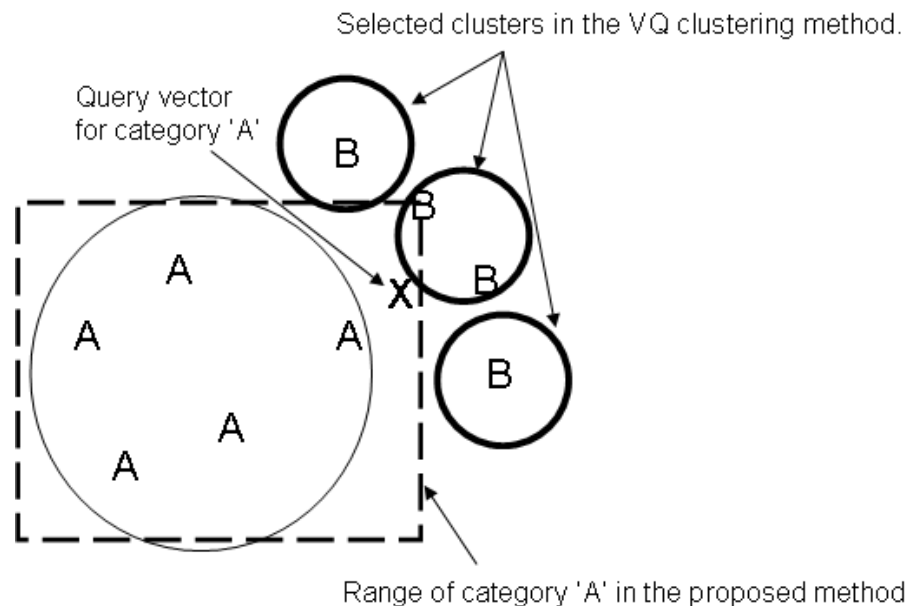


図 3.11 VQ クラスタリング法が選択失敗して、MAH 法が成功する例

図 3.8 に示されるトータルな認識処理の性能をみると、MAH 法が最もよい手法であるといえる。さらに、図 3.9 と図 3.10 によって、MAH 法は、短い処理時間で、高精度な分類精度を維持したまま選択カテゴリ数を相対的に小さくすることができることがわかる。例えば、図 3.8 において分類エラー率として 0.07% 程度を想定すると、MAH 法では、1 文字を 100msec. 未満で認識することができるが、類似の分類エラー率で、他の手法は処理時間が 100msec. を超えてしまう。

次に、大分類のエラーについて分析を行った。各手法を比較するために、全体の認識処理時間が 90msec. 程度になるように各手法のパラメータを選択した。ただし、ANN 法は、全体の認識処理時間が非常に遅いため、全体の認識処理時間と分類エラー率を考慮して決めた。選択したパラメータは、以下の通りである。AM 法では、マージン値を 10 とし、MAH 法では、マージン値を 20 とした。また、VQ クラスタリング法では、クラス数を 100 とした。ANN 法では、 eps を 3.0、 k を 1000 に設定し、LSH 法では、 L が 20、 k が 5、 w は 20 に設定した。

提案手法による改良部分を分析するために、提案手法が正しく分類できて、かつ、比較手法で分類失敗する場合を調べた。VQ クラスタリング法では、クラスは入力クエリベクトルと各クラス中心の間の距離値によって選択されるので、クラスタの大きさが異なる

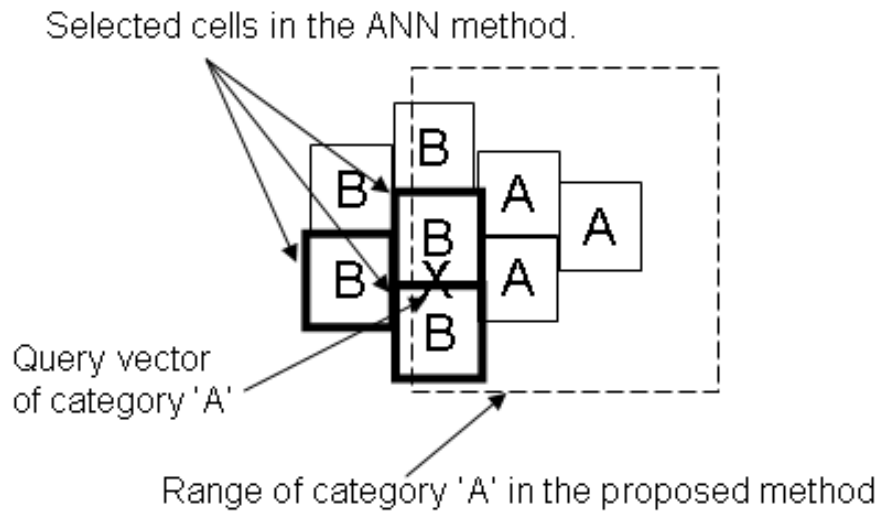


図 3.12 ANN 法が選択失敗して、MAH 法が成功する例

る場合は、入力クエリベクトルに近いより小さなクラスタが選ばれやすくなる。この場合、正しいカテゴリが選択される小さなクラスタに入っていない場合は、VQ クラスタリング法は、大分類に失敗することになる。例えば図 3.11 において、カテゴリ 'A' のクエリベクトルが入力された場合、入力クエリベクトルの近くのクラスタが選択されるが、この選択されたクラスタはカテゴリ 'B' の学習ベクトルだけしか所属していないので、大分類で正しいカテゴリを選択することができない。これに対し、MAH 法ではカテゴリ 'A' の範囲は、学習サンプルのベクトル位置にマージンを加えて設計されているために、正しいカテゴリ 'A' を選択することができる。

ANN 法では、kd-tree 構造辞書の中の各セルは、1つの学習ベクトルによって決定される。そのため、特徴空間の中で、学習サンプルの密度が大きいところではセルの大きさは非常に小さくなる。ANN 法の大分類の処理時間を短くするために、選択するベクトルの数を減らす必要があるが、選択ベクトル数を減らすと、選択されたベクトル集合の中のユニークカテゴリ数は小さくなるため、分類精度が低下してしまう。図 3.12 では、カテゴリ 'A' の入力クエリベクトル近くのセルは、全てカテゴリ 'B' であるので ANN 法では正しく分類ができないが、MAH 法では、クエリベクトルが、学習サンプルとマージン値で求められる 'A' の範囲内にプロットされるため、正しいカテゴリを選択することができる。

LSH 法と MAH 法では、入力クエリベクトルは、各手法で設計されている各軸へ投影され、軸の上にある候補カテゴリが選択される。しかし、LSH 法では、MAH 法とは異

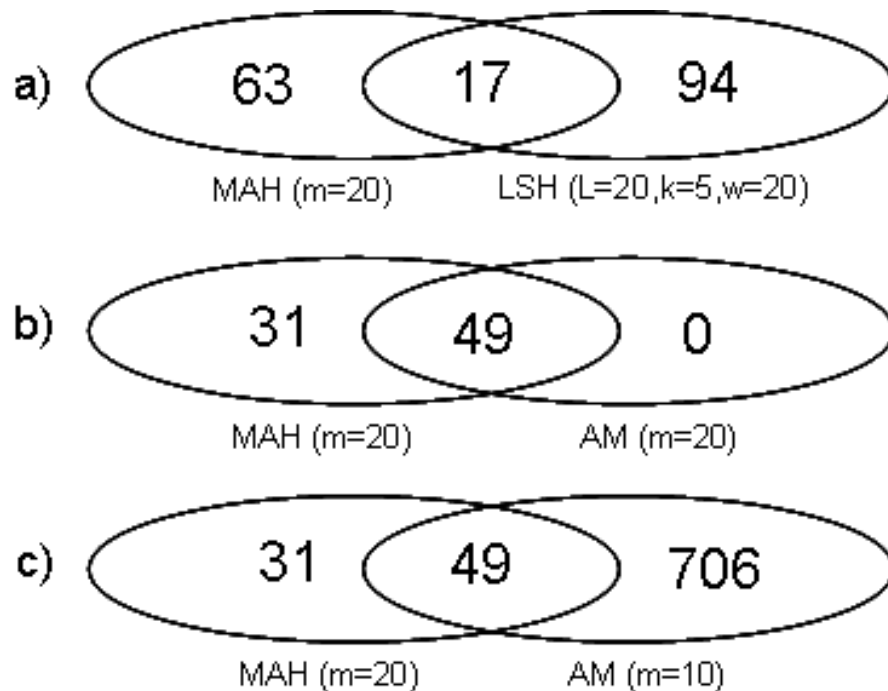


図 3.13 各手法での選択エラー文字数

なるランダムな軸を使用しているために、直接的な比較はできない。LSH 法で大分類が失敗する文字は、LSH 法で使用される全ての軸上のハッシュ bucket の外に投影されている。しかし、その中のいくつかは MAH 法で正しく選択される。ここでは、LSH 法の 0.095% の分類エラー率と、MAH 法の 0.069% の分類エラー率となるパラメータを用いて大分類できなかった文字を比較した。この場合の LSH 法のパラメータは、 $L=20$ 、 $k=5$ 、 $w=20$ であり、MAH 法のパラメータは、マージン値が 20 である。分類エラーの分析結果を図 3.13(a) に示す。MAH 法だけによる大分類エラー文字数は、63 であり、LSH 法だけによる大分類エラー文字数は、94 であった。また、両方の手法でエラーした文字数は 17 であった。この分析で、LSH 法において、エラーした文字の総数 111 の原因は、LSH 法で使用される全ての軸上のハッシュ bucket の外に投影されていることである。また、111 文字の中の 94 文字は、MAH 法によって正しく選択されている。これは、これらの文字は、MAH 法の辞書の中の正しいカテゴリの範囲にプロットされたからである。

図 3.8 において、全体の認識処理時間がおおよそ 90msec. のあたりを見ると、AM 法のマージン値が 10 であるのに対し、MAH 法はマージン値が 20 となっている。この場合、図 3.14 に示す様に、入力クエリベクトルが AM 法の範囲の外で、かつ、MAH 法の範囲の中にプロットされることが可能となる。このような場合によって、AM 法と比較した場合

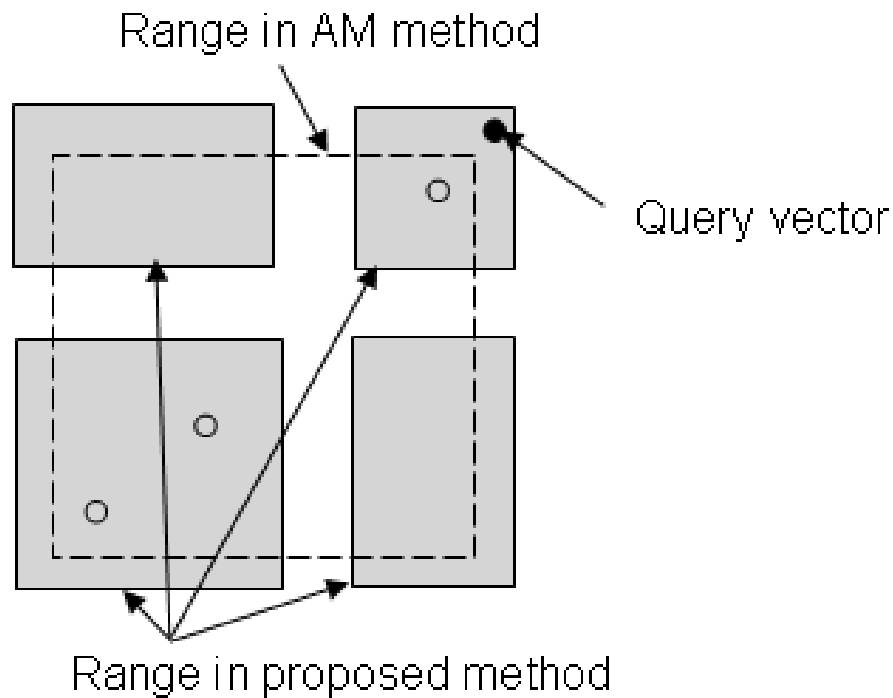


図 3.14 AM 法が選択失敗して、MAH 法が成功する例

の MAH 法の小さな分類エラー率が達成されていると考えられる。しかし、両方の手法で同じマージン値が使用されると、特徴空間の中での AM 法の各カテゴリのカバー範囲は、MAH 法のそれよりも大きくなるために、大分類の分類エラー率は、AM 法の方が小さくなる。この現象を、マージン値 10 と 20 の AM 法と、マージン値が 20 の MAH 法を比較して実験で確かめた。両方の手法でマージン値 20 を使用した場合の結果を図 3.13 の (b) に示す。AM 法だけ選択エラーとなる文字数はゼロである。これは、AM 法の各カテゴリの範囲が MAH 法での各カテゴリの範囲を完全に包含していることを表す。AM 法でマージン値を 10 とした場合の結果は、図 3.13 の (c) に示される。この場合は、AM 法だけ選択エラーとなる文字数は 706 となった。これは、706 のクエリベクトルが AM 法のカテゴリ範囲の外側にプロットされ、かつ、MAH 法のカテゴリ範囲の内側にプロットされたことを表している。

次に、提案手法で選択エラーとなるが、比較手法によっては選択に成功する場合について分析を行った。MAH 法で正しく選択できない文字は、全部で 80 あった。この内、比較対象のどの手法でも正しく選択できない文字は 16 文字であった。これらの文字は、数

表 3.3 比較手法における正しく選択できた文字数

Methods	LSH	AM	VQ Clustering	ANN
<i># of correct selections</i>	20	31	42	51

字、アルファベット、記号のいずれかであった。正しく選択されない理由は、それらの文字から作成された特徴ベクトルが、特徴空間の中で学習サンプルの分布とかけ離れていたからである。その原因は、文字の位置ずれや、イタリック体のような変形、ノイズの付加、文字のかすれであった。**MAH**法で正しく選択されない80文字の中で、これらの文字以外の64文字について、今後**MAH**法をさらに改良できるヒントを得るために、分析を行った。結果を表3.3に示す。この表では、**ANN**法と**VQ**クラスタリング法は、**LSH**法や**AM**法よりも良い結果となっている。この理由は、**kd-tree**構造辞書が特徴空間の全体をカバーするために、**ANN**法では学習サンプルから外れた文字でも正しく選択できるからであると思われる。また、同様にクラスタと入力クエリベクトル間の距離を使うため、**VQ**クラスタリング法も特徴空間全体をカバーした辞書をもっていることから、良い結果が得られたと思われる。**LSH**法や**AM**法と同様に、**MAH**法は軸の上で範囲が決まっているハッシュテーブルを使用する。学習サンプルから外れた文字の場合は、正しいカテゴリのハッシュテーブルの範囲から外れたところへ投影されるため、これらの手法では、このような文字は、正しく選択できない。

この分析結果から、ハッシュ法は、学習サンプルから外れた文字の場合に問題があることがわかった。すなわち、もしハッシュ法が特徴空間を全てカバーできるとしたら、分類エラー率はより低下させることができる。しかし、これは、選択されるカテゴリ数の増大を引き起こし、処理時間の増大につながる。**MAH**法に関しては、もし特徴空間の中であまり学習サンプルベクトルが存在しないところで、より大きなマージン値を使うことができれば、選択されるカテゴリ数をあまり増やさずに分類エラー率を低下させることができると思われる。

最後に、各種法の辞書サイズを比較した。結果を表3.4に示す。**VQ**クラスタリング法は、辞書として、各クラスタの代表ベクトルの情報と、各クラスタに所属するカテゴリの情報だけあればよいので、辞書サイズは、他の手法に比べて非常に小さい。**LSH**法、**AM**法、**MAH**法は、各カテゴリをビット表現して、各軸上のハッシュテーブルの辞書を構成しているので、これらの手法の間でテーブル数や軸の数が異なっても似たような辞書サイ

表 3.4 各手法の辞書サイズ

Methods	LSH	AM	MAH	VQ Clustering	ANN
Size of dictionary in Mbytes	1.4	2.7	2.7	0.05	41.1

ズとなっている。これらの手法では、辞書内の全カテゴリについて記述が必要なために、辞書サイズは、VQ クラスタリング法よりも大きくなる。ANN 法の辞書サイズは、比較対象の手法の中で一番大きかった。これは、全ての学習サンプルのベクトル情報と、特徴空間でのセル間の識別面の情報を辞書に記述する必要があるためである。分類エラー率との関係では、VQ クラスタリング法は、非常に小さな辞書サイズにもかかわらず比較的高い分類精度を確保できるので、メモリ使用量が制限される環境では有利である。

3.5 結論と今後について

本論文では、MAH 法と名付けられた新しい大分類手法を提案した。この手法は、OCR に使用されることを想定して、ハッシュ法をベースとして開発されたものである。MAH 法では、ハッシュテーブル辞書で学習文字画像の各カテゴリの存在範囲を小さくすることにより、従来手法の AM 法の大分類精度を維持したまま、選択されるカテゴリ数を削減することができる。さらに、提案手法を使うことにより、大分類と詳細認識からなる文字認識の処理時間を、従来手法より短くすることができる。本論文では、提案手法と、従来手法の AM 法、VQ クラスタリング法、ANN 法、および LSH 法について、116528 の日本語活字文字を使った評価実験を行い、性能を比較した。その結果、MAH 法は、大分類の分類精度、選択するカテゴリ数の小ささ、および総認識処理時間の短さのバランスにおいて従来手法よりもよい結果となった。

MAH 法では、各学習サンプルに加えるマージン値の選択が重要である。なぜならば、この値は、処理時間と分類精度の両方に影響を与えるからである。今後は、より実用的なアプリケーションを実現するために、各カテゴリの特徴空間の中での分布を考慮して、各学習サンプル毎に最適なマージン値を選択する調査を行っていく予定である。

第4章

省メモリ化のための研究成果

4.1 はじめに

近年、携帯電話やスマートフォンをはじめとする携帯機器が普及してきた。携帯機器は、通信だけでなく、携帯機器に備わっているカメラで画像を撮影することができ、携帯機器で画像を処理するアプリケーションも増えている。画像を撮影するアプリケーションは、様々なものがあるが、その中で、文書や看板などの文字情報を含む画像を扱うアプリケーションでは、画像中の文字を認識することで、画像からテキストを抽出し、それを翻訳やキー入力の代わりとして再利用したり、認識したテキストを画像に付与して、キーワードによって画像を検索させることができるものがある。撮影操作だけで手軽に文字入力や画像検索ができるため、近年、この機能のニーズが広がってきている。

この機能を実現する場合、従来のようなサーバやPCでの文字認識処理ではなく、処理能力が低く、使用メモリ量に制限がある携帯機器の上での文字認識技術が必要である。また、これとは別に、帳票処理業務などを行う大規模なコンピュータシステムにおいて、大量の画像を対象に文字認識するニーズもある。この場合には、プロセッサ等の処理単位を大量に用意して、並列に処理することで効率を上げている。しかし、認識対象の画像規模と比較すれば1つの処理単位のハードウェアリソースは制限されたものになる。このように、文字認識処理の効率化のニーズのためには、制限されたハードウェアリソースでの処理が必要である。

一方、従来の研究によって、高精度な認識を実現するためには、認識辞書内のカテゴリごとの単純な代表ベクトルと入力文字の特徴ベクトルとの間の類似度（相違度）だけを使うよりも、特徴空間でのカテゴリ毎の分布を考慮した2次識別関数による類似度（相違

度)を用いた方がよいことが知られている。このため、高精度認識を達成するために、辞書サイズは、単純な代表ベクトルだけの場合より大幅に大きくなることは必須である。このように、物理的リソースに直接影響する辞書サイズと、文字認識率の間には、トレードオフの関係がある。

この課題を解決するために、文字認識処理において辞書サイズを制限しながら2次識別関数による高精度認識を達成する以下のような手法が考案されている。

- 1) 特徴ベクトルなどの情報の1要素のビット幅を縮小する方法 [119]
- 2) 2次識別関数で使用する固有ベクトルの軸数を制限する手法 [120]
- 3) Long らの Compact MQDF 手法 [93] [94]

これは、2次識別関数で使用する辞書の中で固有ベクトルを小さい固定次元数の部分ベクトルに分割し、全ての部分ベクトルを対象にクラスタリングした結果のクラスタ番号で置き換えることで辞書サイズを削減する手法

しかし、1)の手法は、特徴ベクトルなどの1要素のダイナミックレンジが小さくなるために精度が低下する。2)の手法は、軸数を制限しすぎると分布形状を正しく表現できなくなり、精度が低下する。また、3)の手法は、分布形状を表現する固有ベクトルを近似したベクトルに置き換えるもので、類似カテゴリの識別能力が低下し、精度が低下する。3)の手法については、4.3章で詳しく説明する。

本論文は、3)の手法をベースとして、改良を加えることで、ほとんど精度低下を起こさずに2次識別関数で使用する辞書サイズを削減する手法である。以下、4.2章でMQDF手法を説明し、4.3章でベースとなる Compact MQDF 手法を詳細に説明する。4.4章では本論文で提案する手法を記述し、4.5章で実験結果を示す。4.6章で考察を行い、最後にまとめを記す。

4.2 MQDF 手法

Long らの手法のベースとして、まず MQDF 手法について説明する。認識過程において、入力パターンが、最大事後確率となるカテゴリに分類されるというベイズの決定規準に基づいて、各カテゴリが多変数のガウス分布の密度関数を持つという仮定の下で、二次の判別関数(QDF)が得られる。

木村らによって提案された MQDF [121] は、計算効率および分類性能を改善するため

に、K-L 展開と、小さな固有値を一定値にすることによって QDF の修正を行ったものである。ベイズの規則によれば、事後確率は、次式で求められる。

$$P(\omega_i|x) = \frac{P(\omega_i)p(x|\omega_i)}{p(x)}, i = 1, 2, \dots, M \quad (4.1)$$

ここで、 M は、カテゴリの数、 $P(\omega_i)$ は、カテゴリ ω_i の事前確率、 $p(x|\omega_i)$ はカテゴリ ω_i の確率密度関数、 $p(x)$ は混合確率密度関数である。 $p(x)$ は、カテゴリに依存しないため、式 4.1 の分子部分は、式 4.2 のように判別関数として分類に使用することができる。

$$g(x, \omega_i) = P(\omega_i)p(x|\omega_i), i = 1, 2, \dots, M \quad (4.2)$$

各カテゴリの確率密度関数が多変数ガウス関数であると仮定すると、カテゴリ ω_i の確率密度関数は、以下のようになる。

$$p(x|\omega_i) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left[-\frac{(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)}{2} \right] \quad (4.3)$$

ここで、 μ_i と Σ_i は、それぞれ、カテゴリ ω_i の平均ベクトルと共分散行列を表す。 D は、 μ_i の次元数である。

式 4.3 を式 4.2 に代入し、対数を取り、等しい事前確率の共通項を省略すると次式の QDF が導かれる。

$$g_0(x, \omega_i) = (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) + \log |\Sigma_i| \quad (4.4)$$

QDF は、最小評価値のカテゴリを入力パターンに割り当てる意味で、距離指標として使うことができる。K-L 変換によって共分散行列は、次式のように対角行列を使って変形することができる。

$$\Sigma_i = \Phi_i \Lambda_i \Phi_i^T \quad (4.5)$$

ここで、 $\lambda_{ij}, j=1, 2, \dots, D$ は、カテゴリ ω_i の共分散行列 Σ_i の固有値で、大きさの降順に並べたもの。 $\Lambda_i = \text{diag} [\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{iD}]$ は、固有値を対角成分とする行列。 $\phi_{ij}, j=1, 2, \dots, D$ は、カテゴリ ω_i の共分散行列 Σ_i の固有ベクトルで、固有値 λ_{ij} に対応する

ように並べたもの。 $\Phi_i = [\phi_{i1}, \phi_{i2}, \dots, \phi_{iD}]$ は、カテゴリ ω_i の固有ベクトル ϕ_{ij} を順番に並べた固有ベクトル行列である。また、 Φ_i は正規直交ベクトルで、 $\Phi_i^T \Phi_i = I$ である。式 4.5 に従えば、QDF は、固有値と固有ベクトルの形式で下のよう書き直すことができる。

$$\begin{aligned} g_0(x, \omega_i) &= \left[\Phi_i^T (x - \mu_i) \right]^T \Lambda_i^{-1} \Phi_i^T (x - \mu_i) + \log |\Lambda_i| \\ &= \sum_{j=1}^D \frac{1}{\lambda_{ij}} \left[\phi_{ij}^T (x - \mu_i) \right]^2 + \sum_{j=1}^D \log \lambda_{ij} \end{aligned} \quad (4.6)$$

小さな固有値を定数 δ_i で置き換えることにより、MQDF が下の式のように求まる。

$$\begin{aligned} g_1(x, \omega_i) &= \sum_{j=1}^K \frac{1}{\lambda_{ij}} \left[\phi_{ij}^T (x - \mu_i) \right]^2 + \sum_{j=K+1}^D \frac{1}{\delta_i} \left[\phi_{ij}^T (x - \mu_i) \right]^2 + \sum_{j=1}^K \log \lambda_{ij} + (D - K) \log \delta_i \\ &= \frac{1}{\delta_i} \left[\|x - \mu_i\|^2 - \sum_{j=1}^K \left(1 - \frac{\delta_i}{\lambda_{ij}} \right) \left[\phi_{ij}^T (x - \mu_i) \right]^2 \right] + \sum_{j=1}^K \log \lambda_{ij} + (D - K) \log \delta_i \end{aligned} \quad (4.7)$$

ここで、 K は、主要な固有ベクトルの数を表す。上の式は、下の式で表されるユークリッド距離の不変性を利用している。

$$d_E(x, \omega_i) = \|x - \mu_i\|^2 = \sum_{j=1}^D \left[\phi_{ij}^T (x - \mu_i) \right]^2 \quad (4.8)$$

パラメータ δ_i は、木村ら [121] が提案したようにカテゴリに非依存の定数とすることができる。または、小さな固有値の平均値として計算し、カテゴリに依存する定数とすることもできる。

4.3 Compact MQDF 手法

MQDF では、各カテゴリにおいて学習用サンプルで学習した、平均ベクトル μ_i 、共分散行列 Σ_i の上位 K 個の固有値 λ_i 、固有ベクトル ϕ_i などのベクトル情報を辞書として用意する必要がある (図 4.1)。MQDF 辞書の大きさは、1つのカテゴリにおいて、カテゴリのインデックスまたは文字コードを表す1次元の要素の他に、 μ_i が D 次元、 λ_i が K 次元、 ϕ_i が $K \times D$ 次元が必要である。この内、MQDF 辞書の中で圧倒的な部分を占める

Category	eigen value	eigen vector			
C_1	a_{11}	v_{11}^1	v_{11}^2	...	v_{11}^D
	a_{12}	v_{12}^1	v_{12}^2	...	v_{12}^D
	:	:	:	...	:
	a_{1K}	v_{1K}^1	v_{1K}^2	...	v_{1K}^D
C_2	a_{21}	v_{21}^1	v_{21}^2	...	v_{21}^D
	a_{22}	v_{22}^1	v_{22}^2	...	v_{22}^D
	:	:	:	...	:
	a_{2K}	v_{2K}^1	v_{2K}^2	...	v_{2K}^D
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
C_M	a_{M1}	v_{M1}^1	v_{M1}^2	...	v_{M1}^D
	a_{M2}	v_{M2}^1	v_{M2}^2	...	v_{M2}^D
	:	:	:	...	:
	a_{MK}	v_{MK}^1	v_{MK}^2	...	v_{MK}^D

図 4.1 MQDF 辞書

のが固有ベクトル部分である。カテゴリ数を M 、平均ベクトルの次元数を D 、使用する上位の固有値固有ベクトルの数を K 個、各ベクトルの 1 つの要素を 4 バイトの浮動小数点とすると、固有ベクトル部分のサイズは、 $D \times K \times M \times 4$ となる。例えば、 $D=288$ 、 $K=10$ 、 $M=3700$ の場合は、42,624,000 バイト（約 43M バイト）となってしまう。

Long らは、各カテゴリの固有ベクトル部分を圧縮するために、VQ クラスタリング手法の一種を使用した。すなわち、先ず初めに、 D 次元固有ベクトル ϕ_{ij} を、同じ大きさの Q 個の部分ベクトルに分割し、 D_Q 次元のサブベクトル $\phi_{ij1}, \phi_{ij2}, \dots, \phi_{ijQ}$ を生成する。ここで $D=D_Q \times Q$ という関係となる。

次に、すべてのカテゴリから生成された部分ベクトルを 1 つの集合に入れ、この部分

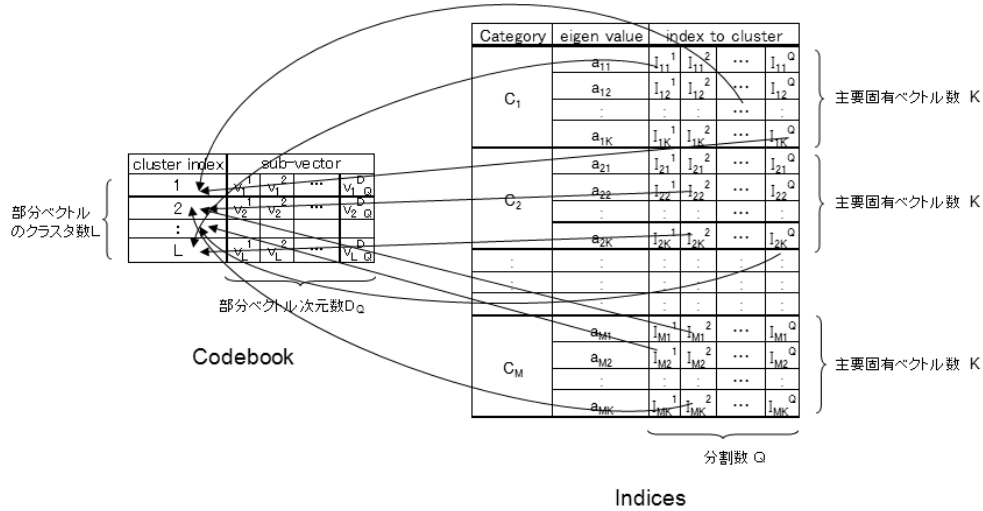


図 4.2 Compact MQDF 法の Codebook と Indices

ベクトルの集合を対象とした **LBG** クラスタリング [122] を行う。その結果、 L 個のクラスが作成され、クラスタリング前の部分ベクトル ϕ_{ijq} , ($i=1, \dots, M, j=1, \dots, K, q=1, \dots, Q$) は、小さな L 個の D_Q 次元の部分ベクトルのプロトタイプにクラスタリングされる。生成された各クラスにはクラス番号を付け、 D_Q 次元の部分ベクトルの代表ベクトルを求めておく。

元の部分空間の固有ベクトルの中の D_Q 次元の部分ベクトルは、その最近傍の D_Q 次元の部分ベクトルのプロトタイプで近似され、 D_Q 次元の部分ベクトルの代わりに最近傍のプロトタイプのインデックス番号で代理表現するようにする。

この仕組みを図 4.2 に示す。図 4.2 は、2 つの部分から構成されている。すなわち **Codebook** と、**Indices** である。

Codebook は、部分ベクトルのクラスタリング結果の情報であり、 L 個のクラスが、クラスインデックス番号の小さい順に並んで格納されている。1 つのクラスには、クラスに所属する部分ベクトルの代表ベクトルである D_Q 次元のベクトルが保存されている。

Indices は、従来の固有ベクトル行列の代わりの情報である。この中には M 個のカテゴリの固有ベクトル行列がある。1 つの固有ベクトル行列は、 K 個の固有ベクトルからなる。1 つの固有ベクトルは、 Q 次元のベクトルで、その要素は **Codebook** の中にあるクラスのインデックス番号 ($1 \sim L$) である。

D_Q 次元部分ベクトルの 1 つの要素を表現するビット幅を r と定義すると、 L が 2 の

($D_Q \times r$) 乗より小さい場合、元の固有ベクトルの D_Q 次元の部分ベクトルは、($D_Q \times r$) のビット幅より小さいビット幅である 1 つのインデックス番号で置き換えて表現されるため、固有ベクトルのサイズは、圧縮できる (図 4.3)。

Codebook のサイズは、部分ベクトルの 1 要素を浮動小数点とすると、 $4 \times D_Q \times L$ バイトとなる。また、Indices のサイズは、クラスタのインデックス番号の最大値 L は 2 バイトの unsigned short で表現できるため、 $2 \times Q \times K \times M$ バイトとなる。

例えば、 $D=288$, $K=10$, $M=3700$, $D_Q=4$, $Q=72$, $L=500$ の場合は、図 4.3 の Codebook のサイズは、 $4 \times 4 \times 500 = 8\text{K}$ バイト、Indices のサイズは、 $2 \times 72 \times 10 \times 3700 = \text{約 } 5.3\text{M}$ バイトで、Codebook と Indices を合計したサイズでも約 5.3M バイトとなり、元の MQDF 辞書を使用する場合の $1/9$ 程度と大幅に圧縮できることが分かる。

4.4 Hybrid Compact MQDF 手法

Long らの Compact MQDF 手法は、固有ベクトルを、近似したベクトルに変更して、それをインデックス番号で置き換えることにより、精度をほとんど低下させないまま、辞書サイズを大幅に圧縮することが可能である。

一方、日本語のような文字カテゴリ数の多い言語では、辞書内に類似した文字が多数ある。これらを高精度に認識するために、これらのカテゴリの固有値、固有ベクトルの微妙な違いが問題となる。Long らの Compact MQDF 手法は、辞書内の全カテゴリの固有ベクトルを一律にクラスタリング対象とするため、類似文字の固有ベクトルから作成された部分ベクトルが同じクラスタに分類され、結果として、類似文字集合内で、固有ベクトルが完全に一致してしまう可能性がある。この場合、類似文字のカテゴリの認識に使う情報としては、違いは代表ベクトルと固有値だけとなるが、もともと文字の形状が類似しているため、分布も類似していると推定されるので、この 2 つの情報も類似している可能性が高い。その結果、類似文字の識別能力は、オリジナルの辞書を圧縮しない MQDF 手法よりも低下すると推定できる。

この欠点を克服するために、本論文では、Compact MQDF 法を改良して、類似文字の識別能力を維持したまま、辞書サイズを圧縮し高精度を維持する MQDF 手法を提案する。

本手法では、最初にカテゴリ間の類似性を評価して類似カテゴリ集合を作成する。この類似カテゴリ集合に入らないカテゴリは、他のどのカテゴリとも異なっている形状の文字であることになり、このカテゴリだけを対象に Long らの Compact MQDF の処理を行い、近似固有ベクトルを使用した認識を行う。類似カテゴリ集合に所属するカテゴリは、高精

cluster index	sub-vector			
1	v_1^1	v_1^2	...	v_{1Q}^D
2	v_2^1	v_2^2	...	v_{2Q}^D
:				
L	v_L^1	v_L^2	...	v_{LQ}^D

a) Codebook

Category	eigen value	index to cluster			
C_1	a_{11}	I_{11}^1	I_{11}^2	...	I_{11}^Q
	a_{12}	I_{12}^1	I_{12}^2	...	I_{12}^Q
	:	:	:	...	:
	a_{1K}	I_{1K}^1	I_{1K}^2	...	I_{1K}^Q
C_2	a_{21}	I_{21}^1	I_{21}^2	...	I_{21}^Q
	a_{22}	I_{22}^1	I_{22}^2	...	I_{22}^Q
	:	:	:	...	:
	a_{2K}	I_{2K}^1	I_{2K}^2	...	I_{2K}^Q
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
C_M	a_{M1}	I_{M1}^1	I_{M1}^2	...	I_{M1}^Q
	a_{M2}	I_{M2}^1	I_{M2}^2	...	I_{M2}^Q
	:	:	:	...	:
	a_{MK}	I_{MK}^1	I_{MK}^2	...	I_{MK}^Q

b) Indices

図 4.3 Compact MQDF 手法の辞書

度認識のために詳細な識別処理が必要なものであるので、オリジナルの固有ベクトルを使用した MQDF 認識処理を行う。これにより、辞書サイズを押さえながら Compact MQDF より高精度の認識が実現できる。具体的なアルゴリズムを以下に示す。

Compact MQDF 法と同様に、本提案手法も固有ベクトルだけでなく、代表ベクトルや、固有値など他の情報にも適応が可能である。しかし、ここでは簡単のために固有ベクトルだけを対象として説明する。前述したように、MQDF 辞書の 1 つのカテゴリの情報は、図 4.1 に示す様に以下のようなものがある。

- ・カテゴリのインデックス番号、または、文字コード

- ・ カテゴリの代表ベクトル
- ・ 固有値ベクトル
- ・ 固有ベクトル行列

この中のカテゴリの代表ベクトルの情報を使用して、先ず、類似しているカテゴリを1つの集合とする処理を行う。

1) カテゴリ代表ベクトルを使用した類似カテゴリ集合の作成

オリジナルの MQDF 辞書内のカテゴリ代表ベクトルを対象に、階層的クラスタリング [123] を用いて、類似カテゴリ集合を作成する。階層的クラスタリングは、以下のアルゴリズムで行う。

step1: 各カテゴリの代表ベクトルをそれぞれ1つのクラスタとする

step2: すべてのクラスタの代表ベクトル間の距離値を計算する

step3: すべてのクラスタ間の距離値の中で最も小さい距離値となるクラスタペアを求める

step4: 求めた最小距離値がしきい値未満の場合は、クラスタペアを構成する2つのクラスタを1つのクラスタとして統合。クラスタ内に所属するカテゴリのインデックスを記録し、クラスタ重心をクラスタの代表ベクトルとする。最小距離値がしきい値以上の場合は、処理を停止する。

step5: step2～step4 を停止するまで繰り返す

この結果、しきい値を適当に与えると、一部のカテゴリはカテゴリがそのまま1クラスタとなり、他のカテゴリは類似したものが1クラスタに分類され、図 4.4 のようなクラスタが構成される。

2) 類似カテゴリ集合に入らなかったカテゴリを対象とした Compact MQDF 法による辞書圧縮

類似カテゴリ集合に入らなかったカテゴリだけを対象として、Compact MQDF 法による辞書圧縮を行う。すなわち、

a) 固有ベクトルを固定次元数の部分ベクトルに分割

類似カテゴリ集合に入らなかったカテゴリだけを対象として、 D 次元ベクトルである固有ベクトルを Q 個に分割して、各 D_Q 次元の部分ベクトルを作成する。ここで、 $D = D_Q \times Q$ である。作成した部分ベクトルは1つの集合に入れる。

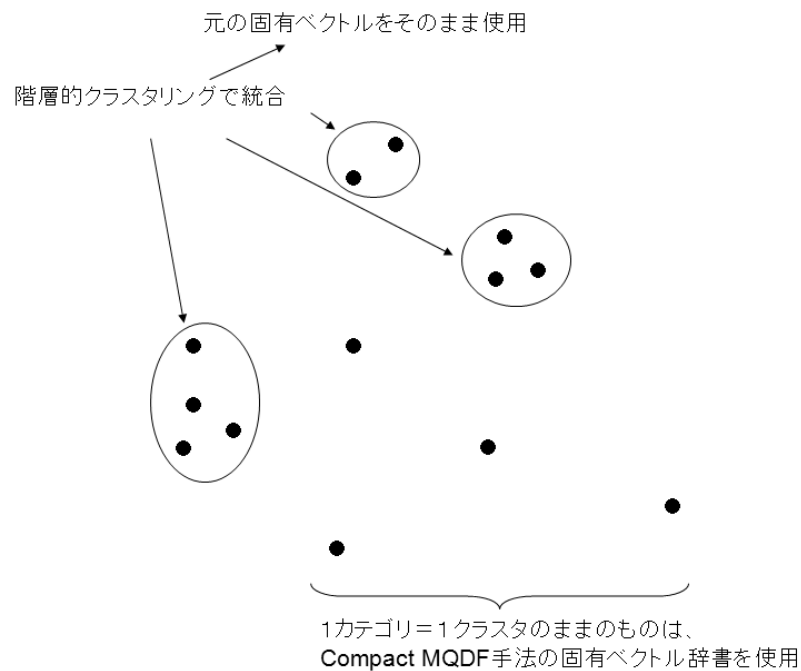


図 4.4 カテゴリ代表ベクトルのクラスタリング結果

b) 部分ベクトルのクラスタリング

部分ベクトルの集合を対象に、クラスタリングを行い、 L 個のクラスタを作成する。本研究ではクラスタリングとして **k-means** 法 [123] を使用した。**k-means** 法クラスタリングの各クラスタの **seed** としては、乱数で発生したベクトルを使用した。また、 L は、パラメータとして与える定数である。

作成された部分ベクトルのクラスタは、クラスタのインデックス番号、1つのクラスタに所属する部分ベクトルのインデックス番号の集合と、クラスタに所属するすべての部分ベクトルから作成された D_Q 次元の代表ベクトルの情報を持つ。

c) Codebook の作成

部分ベクトルのクラスタの中の D_Q 次元の代表ベクトルを、クラスタのインデックス番号順に並べて **Codebook** を作成する

d) Indices の作成

オリジナルの **MQDF** 辞書の1つのカテゴリが、類似カテゴリに入っていない場合、そのカテゴリの固有ベクトルを Q 個に分割し、それぞれ D_Q 次元の部分ベクトルを作成する。この部分ベクトルが所属するクラスタを求め、そのクラスタのインデックス番号で置き換える。この結果、このカテゴリの固有ベクトルは、元の D 次元から、 Q 次元に圧縮

cluster index	sub-vector			
1	v_1^1	v_1^2	...	v_1^D
2	v_2^1	v_2^2	...	v_2^D
:				
L	v_L^1	v_L^2	...	v_L^D

a) Codebook

Category	flag	eigen value	index to cluster or eigen vector					
C_1	0	a_{11}	I_{11}^1	I_{11}^2	...	I_{11}^Q		
		a_{12}	I_{12}^1	I_{12}^2	...	I_{12}^Q		
		:	:	:	...	:		
		a_{1K}	I_{1K}^1	I_{1K}^2	...	I_{1K}^Q		
C_2	1	a_{21}	v_{21}^1	v_{21}^2	...	v_{21}^Q	...	v_{21}^D
		a_{22}	v_{22}^1	v_{22}^2	...	v_{22}^Q	...	v_{22}^D
		:	:	:	...	:	...	:
		a_{2K}	v_{2K}^1	v_{2K}^2	...	v_{2K}^Q	...	v_{2K}^D
:	:	:	:	:		:		
C_i	0	a_{i1}	I_{i1}^1	I_{i1}^2	...	I_{i1}^Q		
		a_{i2}	I_{i2}^1	I_{i2}^2	...	I_{i2}^Q		
		:	:	:	...	:		
		a_{iK}	I_{iK}^1	I_{iK}^2	...	I_{iK}^Q		
:	:	:	:	:		:		
C_j	1	a_{j1}	v_{j1}^1	v_{j1}^2	...	v_{j1}^Q	...	v_{j1}^D
		a_{j2}	v_{j2}^1	v_{j2}^2	...	v_{j2}^Q	...	v_{j2}^D
		:	:	:	...	:	...	:
		a_{jK}	v_{jK}^1	v_{jK}^2	...	v_{jK}^Q	...	v_{jK}^D
:	:	:	:	:		:		
C_M	0	a_{M1}	I_{M1}^1	I_{M1}^2	...	I_{M1}^Q		
		a_{M2}	I_{M2}^1	I_{M2}^2	...	I_{M2}^Q		
		:	:	:	...	:		
		a_{MK}	I_{MK}^1	I_{MK}^2	...	I_{MK}^Q		

b) Indices

図 4.5 提案手法の辞書

される。また、オリジナルの MQDF 辞書の 1 つのカテゴリが、類似カテゴリに入っている場合は、オリジナルの MQDF 辞書の D 次元の固有ベクトルをそのまま使用する。また、各カテゴリが類似カテゴリに所属しているか、していないかの識別フラグを 1 要素分用意する。この方法により作成した固有ベクトルは、図 4.5 のようになる。

本手法では、Codebook のサイズは、部分ベクトルの 1 要素を浮動小数点とすると、 $4 \times D_Q \times L$ バイトとなる。

また、Indices のサイズは、クラスターのインデックス番号の最大値 L は 2 バイトの unsigned short で表現できるため、類似カテゴリに入らないカテゴリでは、1 カテゴリあ

たり $2 \times Q \times K$ バイトとなり、類似カテゴリに入るカテゴリでは、1 カテゴリあたり $4 \times D \times K$ となる。全 M 個のカテゴリの中で、類似カテゴリに入らないカテゴリが N 個 ($0 \leq N < M$) とすると、Indices 全体のサイズは、識別フラグの要素サイズを 1 とし、 $N \times 2 \times Q \times K$ バイト + $(M - N) \times 4 \times D \times K + M$ バイトとなる。 $D=288$, $K=10$, $M=3700$, $D_Q=4$, $Q=72$, $L=500$, $N=1500$ の場合は、提案手法の固有ベクトルのサイズは全体で、 $4 \times 4 \times 500 + 1500 \times 2 \times 72 \times 10 + (3700 - 1500) \times 4 \times 288 \times 10 + 3700 = 27,515,700 \text{ byte} = \text{約 } 28\text{M}$ バイトと、Compress MQDF の辞書よりは大きい、オリジナルの MQDF 辞書の約半分程度に圧縮できる。また、 N の値によって辞書のサイズを制御できることがわかる。

3) Hybrid Compact MQDF 手法での認識処理

文字が入力されると、文字画像から D 次元の特徴ベクトルを抽出する。この特徴ベクトルを使用して MQDF の計算を行う。1 つのカテゴリの計算は、オリジナルの MQDF では、固有ベクトルは各カテゴリの辞書に保存されているものをそのまま使用するが、提案手法の Hybrid Compact MQDF では、 D 次元の合成固有ベクトル領域 ϕ を用意しておき、その場で固有ベクトルを合成しながら計算を進める (図 4.6 を参照)。提案手法の Hybrid Compact MQDF では、先ず識別フラグを調べ、そこが ON の場合は、このカテゴリは類似カテゴリ集合に入っているものである、オリジナル MQDF と同じく、辞書に保存されている固有ベクトルをそのままを使用して合成固有ベクトル領域 ϕ にコピーする。

識別フラグが OFF の場合は、このカテゴリは類似カテゴリ集合に入っていないものである、固有ベクトルの場所にある Q 次元のインデックス番号のベクトルの各要素を参照して、合成固有ベクトル領域 ϕ を合成する。

具体的には、 Q 次元のインデックス番号のベクトルの第 i 目の要素の値が I_i の場合、これを Codebook のインデックスとして使用して、Codebook の第 I_i 番目の D_Q 次元の部分ベクトルを参照し、これを、 D 次元の合成固有ベクトル領域 ϕ の先頭から第 $(I_i - 1) \times D_Q$ 要素目から D_Q 次元分だけコピーする。この処理を Q 個のインデックス番号全てについて繰り返すと、 D 次元の固有ベクトル用の領域 ϕ がすべて埋められる。この後、合成固有ベクトル領域 ϕ を使用して MQDF の計算を行う。

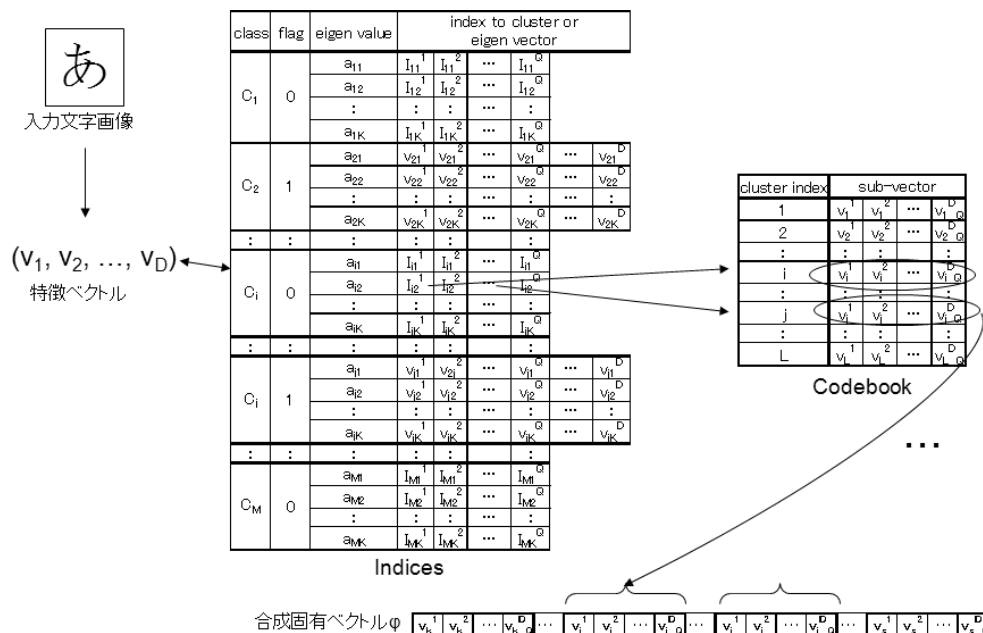


図 4.6 提案手法の認識処理

4.5 評価実験

本手法の有効性を調べるために、辞書サイズと認識精度の関係について、評価実験を行った。使用した特徴ベクトルは、文字画像の輪郭線の画素毎の方向（水平方向からの 0、45、90、135 度の 4 方向）を求め、画像を小さなブロックに分割した時の小ブロック内に入る方向毎の画素頻度から求めたものを使用した [116]。特徴ベクトルの 1 要素のサイズは 1 バイト、次元数は 288 である。

今回の評価実験では、固有ベクトルの要素サイズについては、浮動小数点の数値に大きな定数をかけて、固定小数点化したので、1 要素のサイズは 2 バイトとした。また、カテゴリ数 M は 4300 である。また、辞書サイズとしては、辞書の固有ベクトル部分だけでなく、Codebook、Indices など全部のサイズを合計したものを使用した。

4.5.1 hybrid Compact MQDF 辞書のパラメータ

パラメータとして、以下のものを使用した。

- 1) 類似カテゴリのクラスタリングにおけるクラスタ間の距離 thd

類似カテゴリのクラスタを作成する時の階層的クラスタリングのしきい値として、クラスタ間距離を用いる。階層的クラスタリングは、各クラスタの代表ベクトル間の距離の中で最も小さいクラスタペアを統合していく処理であるが、クラスタ間距離が、しきい値 thd を超える場合に、クラスタリング処理を停止するようにした。 thd をゼロにすると、全てのカテゴリで1カテゴリが1クラスタのままで類似カテゴリがないことになり、即ち、Long らの Compact MQDF 法と完全に一致する。

2) 固有ベクトルを分割してできる部分ベクトルの次元数 D_Q

本手法では、類似カテゴリが無いカテゴリについて、オリジナルの MQDF 辞書の D 次元の固有ベクトルを Q 個に分割して、小さな次元の部分ベクトルを作成する。このときの部分ベクトルの次元数を D_Q とする

3) 部分ベクトルをクラスタリングしてできるクラスタの数 L

本手法では、類似カテゴリが無いカテゴリについて、オリジナルの MQDF 辞書の D 次元の固有ベクトルを Q 個に分割してできた小さな D_Q 次元の部分ベクトルを対象に、クラスタリングを行う。この時に作成するクラスタ数を L とする。

4.5.2 学習用文字

本研究では、124 種類のフォントから各フォントで1文字毎の2値画像を作成する。その後、文字変形をシミュレートするために、1つの文字画像をガウスフィルタでぼかした後に、複数の grayscale しきい値で2値化して、1つの文字画像から多数の2値文字画像を作成する。本研究では、4300 カテゴリ、693~4092 文字／カテゴリ、総数 14,702,094 文字の文字画像を作成した。この文字画像を使用して、先ず、通常の MQDF 辞書を作成する。

4.5.3 評価対象文字

評価用文字として、2種類で実験を行った。データセット1は、学習用文字と同様な手法で作成した。1つの文字画像をガウスフィルタでぼかした後に、学習文字と別な2値化しきい値を使用して2値化してできた4300 カテゴリ、1カテゴリについて21文字~124文字、総数 445518 文字のもの。

データセット2は、学習用の文字とは別に、実際の新聞雑誌など70種類の文書をスキャ

ンして、1文字ずつ切り出して作成した文字画像である。データセット2の中の文字のカテゴリには偏りがあり、辞書内の全てのカテゴリが出現するわけではない。データセット2の中の文字総数は、116,528文字である。

4.5.4 比較実験

本手法の有効性を示すために、ユークリッド距離値を使用した認識手法と、オリジナルのMQDF手法、LongらのCompact MQDF手法の評価を行った。

4.5.5 D_Q と L を固定した場合の評価

まず、固有ベクトルを分割してできる部分ベクトルの次元数 D_Q と、部分ベクトルをクラスタリングしてできるクラスタの数 L を固定して、類似カテゴリのクラスタリングにおけるクラスタ間の距離 thd を変化させた場合について実験した。

データセット1についての結果を図4.7に示す。また、同様に、データセット2についての結果を図4.8に示す。

図4.7、図4.8では、横軸は辞書サイズ数であり、縦軸は文字認識率を示す。また、各図において、Euclid Dist. と記してあるものは、ユークリッド距離値を使用した文字認識結果である。また、MQDF と記してあるものは、オリジナルのMQDF手法を使用した文字認識結果、Compact MQDF と記してあるものは、LongらのCompact MQDF手法を使用した文字認識結果である。そのほかの各点は、本提案手法でパラメータを変化させた場合の認識結果を表す。

4.5.6 thd と L を固定した場合の評価

提案手法において、類似カテゴリのクラスタリングにおけるクラスタ間の距離 thd と、部分ベクトルをクラスタリングしてできるクラスタの数 L を固定して、固有ベクトルを分割してできる部分ベクトルの次元数 D_Q を変化させた場合について実験した。データセット1についての結果を図4.9に示す。また、同様に、データセット2についての結果を図4.10に示す。

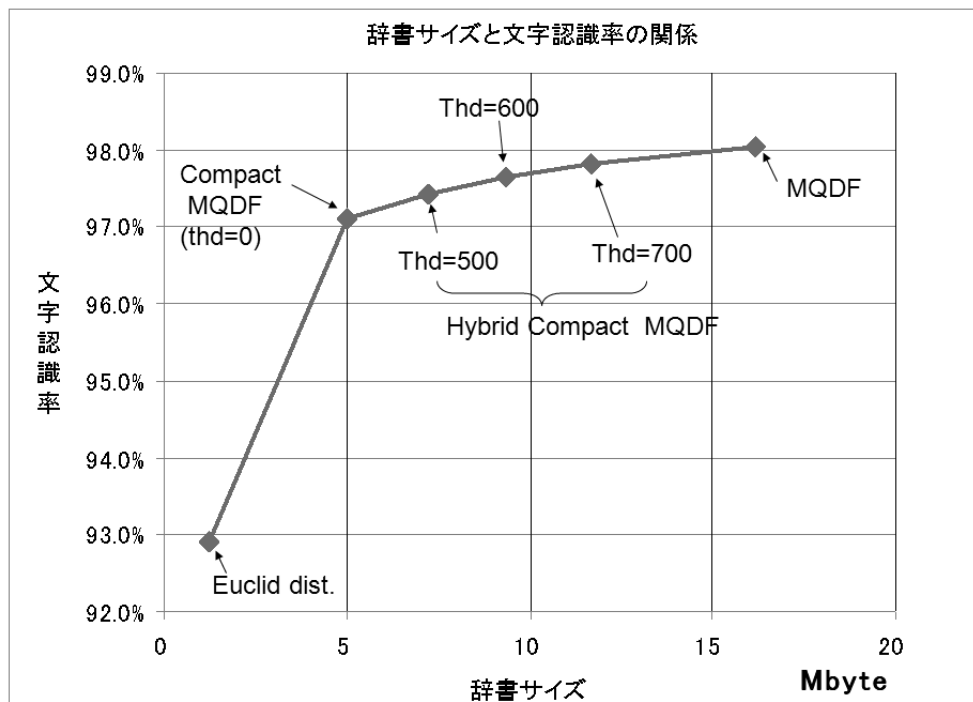


図 4.7 データセット 1 の評価結果, $D_Q=4$, $L=50$ Fix.

4.5.7 thd と D_Q を固定した場合の評価

提案手法において、類似カテゴリのクラスタリングにおけるクラスタ間の距離 thd と、固有ベクトルを分割してできる部分ベクトルの次元数 D_Q を固定して、部分ベクトルをクラスタリングしてできるクラスタの数 L を変化させた場合について実験した。データセット 1 についての結果を図 4.11 に示す。また、同様に、データセット 2 についての結果を図 4.12 に示す。

4.6 考察

4.6.1 D_Q と L を固定した場合

データセット 1 の結果の図 4.7 を見ると、オリジナルの MQDF 手法は、認識精度は、各手法の中で最も高いが、辞書サイズも最大となっている。逆に、Euclid 距離手法は、辞書サイズが最小であるが、認識精度も最小となっている。

Long らの Compact MQDF 手法は、上の 2 つの認識手法の中間の精度、辞書サイズをと

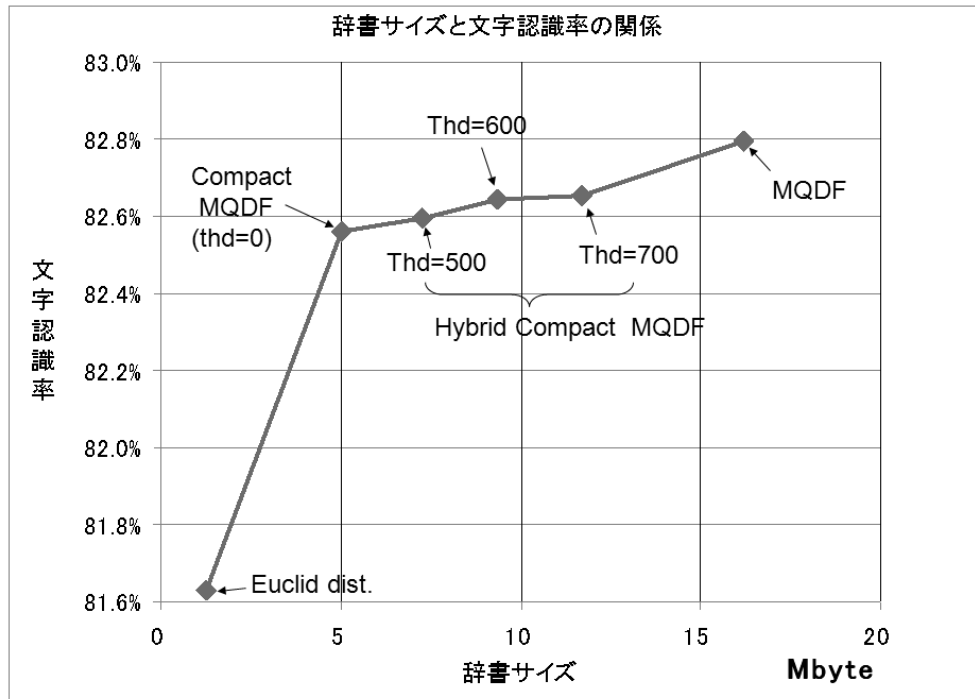


図 4.8 データセット 2 の評価結果, $D_Q=4$, $L=50$ Fix.

なる。しかし、ユークリッド距離法とオリジナル MQDF 手法を直線で結んだ場合の中点の位置にあるのではなく、それよりも上にあるので、Compact MQDF 手法は、文字認識精度をオリジナル MQDF 手法からあまり低下させずにオリジナル MQDF 手法よりも辞書サイズを圧縮できる良い手法であるといえる。次に、本手法の結果は、Compact MQDF 手法とオリジナル MQDF 手法の間にある点として表される。このことから、本手法は、Compact MQDF 手法とオリジナル MQDF 手法の間を埋める手法であるといえ、Compact MQDF 手法と同様に高精度を維持しながら辞書サイズを圧縮できる手法であるといえる。

また、この図は、部分ベクトルの次元数 D_Q と、部分ベクトルのクラスタ数 L を固定して、類似カテゴリを定義する階層的クラスタリングの距離のしきい値 thd を変動させた結果を表している。しきい値 thd を小さくすると、類似カテゴリを作る階層的クラスタリングは近いカテゴリだけを統合して停止するので、ほとんど全てのカテゴリの固有ベクトルが Compact MQDF 法の対象となるため、辞書サイズと認識精度も Compact MQDF 手法に近づく。逆に、しきい値 thd を大きくすると、類似カテゴリを作る階層的クラスタリングは遠いカテゴリまでを統合することになり、ほとんど全てのカテゴリが類似カテゴリとなる。その結果、Compact MQDF 法の対象となる固有ベクトルは少数になるため、辞

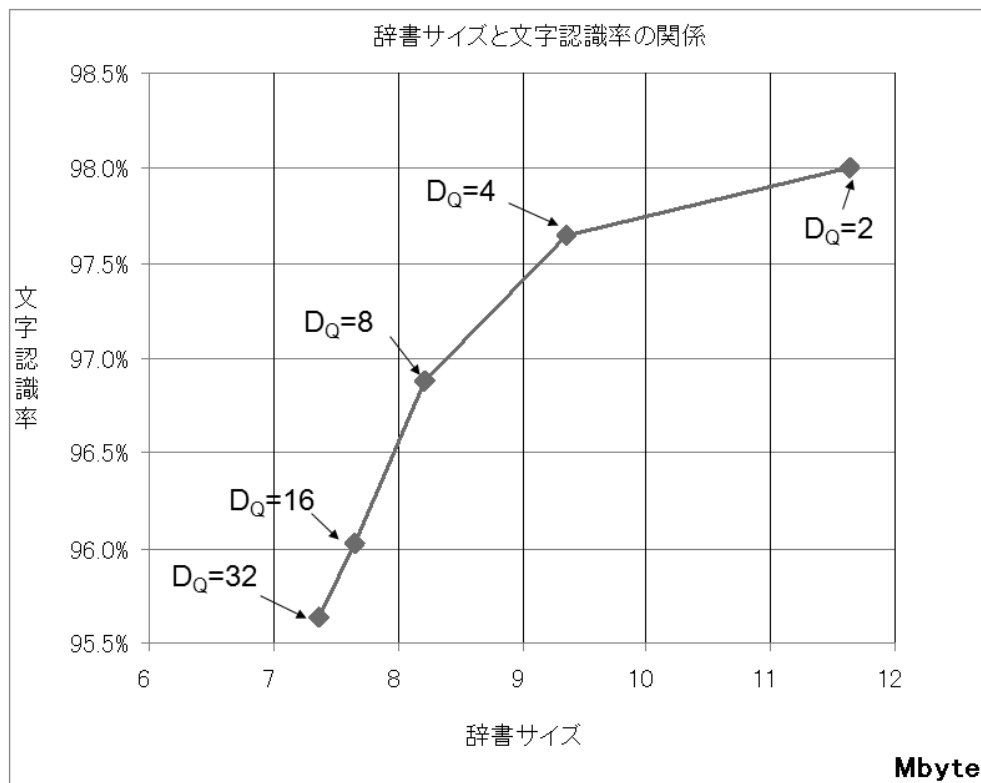


図 4.9 データセット 1 の評価結果, $thd=600$, $L=50$ Fix.

書サイズと認識精度は、オリジナル MQDF 法に近づく。

Compact MQDF 手法との比較では、同じ部分ベクトルの次元数 D_Q 、部分ベクトルのクラスタの数 L において、提案手法の方が辞書サイズは少し大きくなるが、認識精度は向上する。

本手法で thd のパラメータを変動した場合の結果の点群を見ると、Compact MQDF 手法の点とオリジナル MQDF 手法の点を直線で結んだ場合よりも上側にプロットされている。これは、ユークリッド距離法とオリジナル MQDF 手法に対する Compact MQDF 手法の関係と同じで、提案手法は、文字認識精度をオリジナル MQDF 手法からあまり低下させずに Compact MQDF 手法よりも高精度認識を実現できる手法であるといえる。

データセット 2 の結果の図 4.8 をみると、データセット 1 に比べて全体的に認識精度が低下していることがわかる。この原因は、データセット 2 の文字画像は、実際の文書から切り出してきた文字画像であるので、データセット 1 と異なり、学習データと形状が異なるフォントの文字が多数あり、また、データセット 2 の中の文字カテゴリの存在確率に大きな偏りがあるためである。

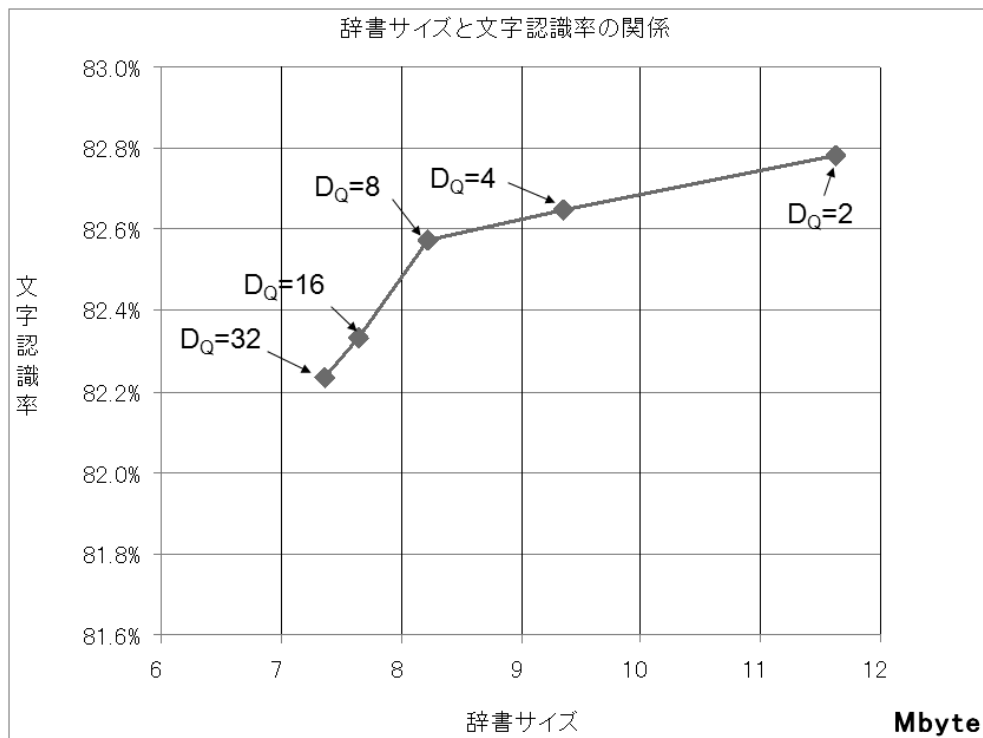


図 4.10 データセット 2 の評価結果, $thd=600$, $L=50$ Fix.

データセット 2 の結果も、データセット 1 の結果と同様で、提案手法は、文字認識精度をオリジナル MQDF 手法からあまり低下させずに Compact MQDF 手法よりも高精度認識を実現できることを示している。

4.6.2 thd と L を固定した場合

図 4.9、4.10 をみると、 D_Q が大きくなるに従って精度が低下しているのがわかる。これは以下の理由による。 D_Q の次元数が大きくなると、部分ベクトルの全ての要素が同時に類似することが難しくなる。そのため、部分ベクトルをクラスタリングして、固定数のクラスタを作成する場合に、実際には類似していない部分ベクトルが 1 つのクラスタに所属する場合が増える。その結果、クラスタ代表の部分ベクトルを作成するときに大きな量子化を行うことになり、代表ベクトルは、そのクラスタを構成する部分ベクトルからかけ離れたものになる。その結果、MQDF の計算で使用する合成固有ベクトルは、オリジナルの固有ベクトルと比べて大きな量子化誤差が生じ、認識精度が低下する。

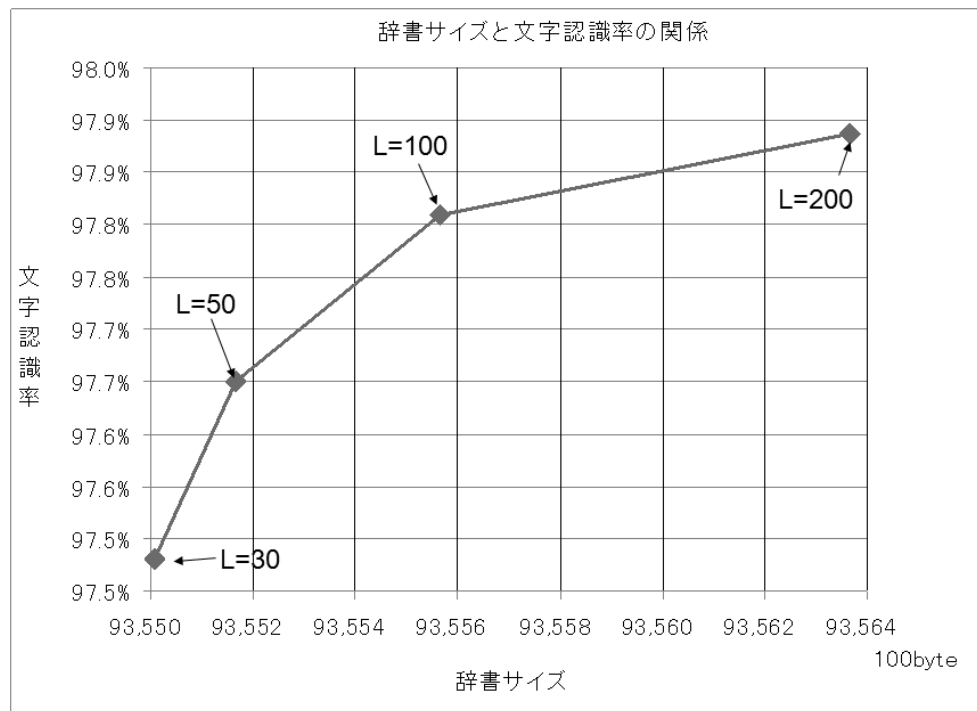


図 4.11 データセット 1 の評価結果, $thd=600$, $D_Q=4$ Fix.

4.6.3 thd と D_Q を固定した場合

図 4.11, 4.12 をみると、クラスタ数 L が小さくなるに従って精度が低下しているのがわかる。これは以下の理由による。クラスタ数 L が小さくなると、クラスタに所属する部分ベクトルの数が大きくなるため、代表ベクトルの量子化誤差が大きくなる。その結果、MQDF の計算で使用する合成固有ベクトルは、オリジナルの固有ベクトルと比べて大きな量子化誤差が生じ、認識精度が低下する。逆に、クラスタ数 L が大きくなると、クラスタに所属する部分ベクトルの数が小さくなるため、代表ベクトルの量子化誤差が小さくなる。その結果、MQDF の計算で使用する合成固有ベクトルは、オリジナルの固有ベクトルに類似したものになるため、認識精度は、オリジナル MQDF に近づく。

また、図 4.11, 4.12 をみると、パラメータ L の変動では、辞書サイズはほとんど変化しないことがわかる。これは、本手法の辞書の中で、Indices に比べて Codebook のサイズの方がはるかに小さいからである。この特性を利用すれば、クラスタ数 L をさらに増やせば、オリジナル MQDF に近づけることができると想像できる。この意味で、現在は、1 つのクラスタに入る類似部分ベクトルをクラスタリングの代表ベクトルで代表させる

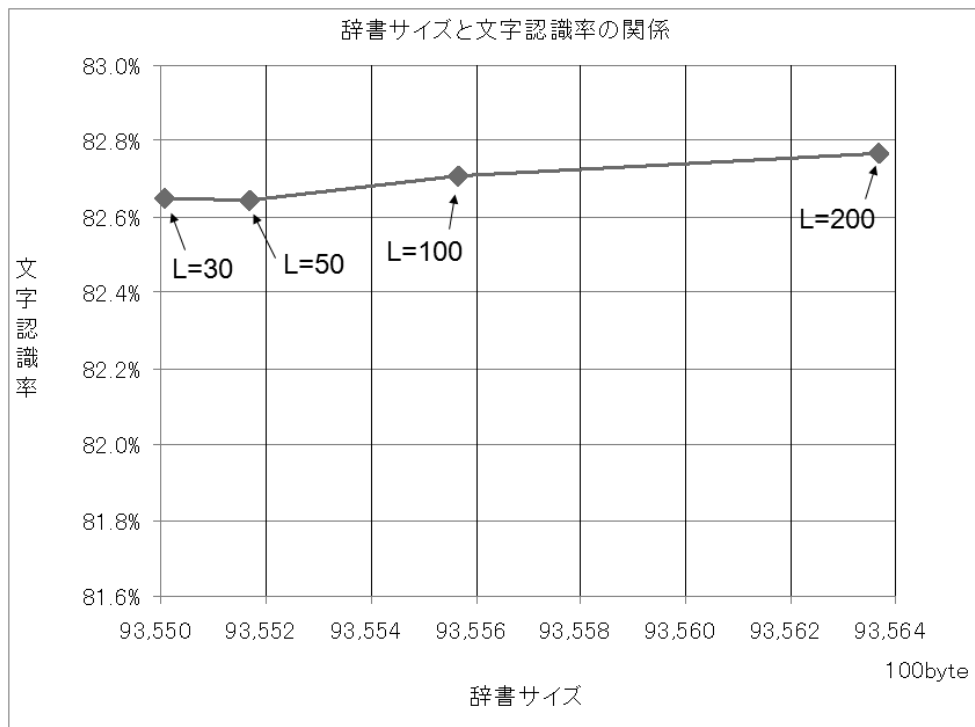


図 4.12 データセット 2 の評価結果, $thd=600$, $D_Q=4$ Fix.

ことで近似しているが、完全に同じ部分ベクトルだけを 1 つのクラスタに入れるような lossless 圧縮を使う事で、オリジナル MQDF と完全に同じ精度で、辞書サイズを小さくすることができる と推察される。

4.7 結論

固有ベクトルを、近似したベクトルに変更して、それをインデックス番号で置き換えることにより、精度をほとんど低下させないまま、辞書サイズを大幅に圧縮することが可能である Compact MQDF 手法をベースとして、Compact MQDF 手法とオリジナル MQDF 手法の中間の性能となる新しい Hybrid Compact MQDF 法を提案した。提案手法は、先ずオリジナルの MQDF 辞書の中の各カテゴリの代表ベクトルを対象とした階層的クラスタリングを行い、一定距離値に収まる類似カテゴリのクラスタを作成する。この類似カテゴリのクラスタに所属するカテゴリは精密な認識のために固有ベクトルはオリジナルのままとし、類似カテゴリクラスタに所属しない他のカテゴリとは離れているカテゴリだけを対象に Compact MQDF の手法でオリジナル固有ベクトルを近似したものに置き換えて、小

さいサイズの MQDF 辞書を作成する。この辞書を使用した Hybrid Compact MQDF 認識手法を評価したところ、Compact MQDF 手法の辞書サイズとオリジナル MQDF 手法の辞書サイズの間サイズの辞書サイズで、Compact MQDF 手法よりも高精度認識が実現できることを確認した。

今後は、固有ベクトルだけでなく、固有値や代表ベクトルなども近似の対象として、さらに省メモリ化を進めることを検討している。また、クラスタリング結果の代表ベクトルで置き換える lossy な近似だけでなく、完全一致するベクトルを1つのクラスターで代表する lossless の圧縮処理との比較や、一部分に lossless 圧縮処理を使用することも行う予定である。

第 5 章

作成した認識技術を使用して映像中のテロップ認識への応用結果

5.1 はじめに

近年、TV のチャンネル化、インターネット利用の増大、また、安価で大容量のビデオレコーダーの普及により、テレビ放送、インターネットおよびイントラネット経由によって収集されるビデオが大量に録画されるようになった。これらの大量に蓄積されたビデオデータを効率的に検索するニーズは、非常に重要で、緊急性のあるものになってきている。

大量のビデオデータへの効率的な検索方法として、テロップ文字情報を使用する方法は有効である。すなわち、あらかじめビデオデータ中のテロップに対して文字認識を行い、認識結果テキストをメタデータとしてビデオと共に保存することで、キーワードによってビデオを検索することができるようになる。一方、TV 番組やビデオの増加や多様化と共に、様々な形態の多くのテロップが使われ始めた。特に、テロップ文字色を変えることによる強調は、日本のテレビ番組の中でよく使用される。

テロップ文字認識を実現するにあたり、まず、ビデオから抽出した各フレーム画像のテロップ領域を抽出することが必要である。本研究では、テロップ領域抽出技術として Gui ら [124] の技術を使用した。テロップは、フレーム画像上の様々な位置に表示されるので、テロップの背景には一般に多様な色から構成される複雑な模様がある。テロップ領域抽出の後、テロップ文字認識を行うために、テロップの文字ストローク画像を抽出しなければならないが、一般的なストローク画像抽出手法 [125] [126] [127] [128] [129] [130] [131] は、テロップ文字列が単一文字色であることを仮定している。

単一色のテロップ文字列からテロップのストローク 2 値画像を生成する方法は、様々な方法がある。最も一般的な方式では、入力カラー画像を、輝度または色情報の使用により複数の 2 値画像レイヤーへ色分解し、その後、1つのレイヤーが文字があるレイヤーとして選択され、そのレイヤーの 2 値画像を使用して文字認識を行うものである。この手法に基づくいくつかの方法 [125] [126] [127] [128] [129] [130] は、テロップ領域の中で、テロップ文字が最も明るい、または、最も暗い画素であると仮定している。

三田ら [125] は、テロップ文字の輪郭領域で輝度のヒストグラムを作り、その後、輝度ヒストグラムで最も明るいピークをテロップ文字の輝度だとして選択する。また、テロップ文字ストローク画像は、テロップの grayscale 画像の中で、ピーク輝度の周囲の輝度範囲の値を使用して抽出される。Chang ら [126] は、大局的 2 値化手法と局所的 2 値化手法を組み合わせた 2 値化手法を使用する。その後、しきい値より低い画素値が、テロップ文字ストローク画素と見なされ、抽出される。Lyu ら [129] は、2 値化の前に、テロップ領域においての文字ストロークの輝度の極性判定を行う。次に、極性判定結果によって明るい、または、暗い画素値を文字ストロークとして抽出し、文字ストローク 2 値画像を作成する。しかし、この手法は、文字単位ではなくテロップ領域単位の極性判定を行うので、テロップ文字が 1 色の場合だけしか正しく処理することができない。

また、いくつかの方法は単語領域検出と、単語領域単位の局所的 2 値化を使用する。Hua ら [131] は、濃淡の極性判定無しにこのタイプの方法を使用した。平均フレーム画像を作成した後、単語領域が抽出される。その後、各単語領域を適応的 2 値化し、最終的に、適応的な 2 値化しきい値より大きな画素値の画素が抽出され、文字ストローク画像が作成される。この手法では、単語領域の中で文字ストロークは最も輝度が大きいことを仮定しているので、1 行に多色文字があるテロップは認識できない。単語領域で適応的 2 値化の後に極性判定を行うことで、多色テロップ文字から文字ストローク画像を正しく抽出することができる可能性があるが、日本語では、単語領域を正しく抽出することが難しいので、この手法は日本語のテロップに適応することは難しい。

上に記述したように、複数色のテロップの場合、これらの方法では、ストローク画像を正しく抽出することができない。これまでの手法と別に、最近、いくつかの方法が、多色テロップ文字認識に対応するようになってきた。

Wakahara ら [132] は、最初に、カラーのテロップ領域画像を、色空間の中での K-means クラスタリングを使用して複数のレイヤーに分解する。その後、各レイヤー 2 値画像の中で黒画素連結領域が求められ、それらの大きさと縦横比によって文字候補が抽出される。

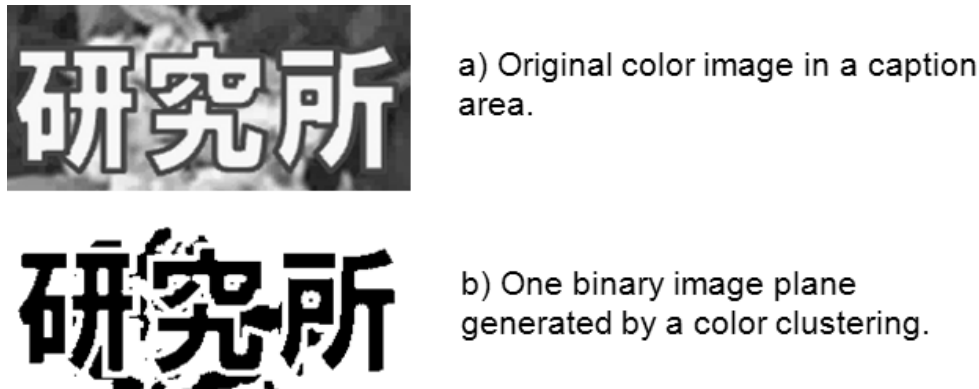


図 5.1 SVM手法で文字ストローク画像抽出に失敗する例

次に **SVM** によって文字とみなされた黒画素連結領域が選択され、領域内文字ストローク画像が抽出される。この手法では、各色分解後のレイヤーに所属する黒画素連結領域が同じ色として扱われるために、多色テロップから文字ストローク画像を正しく抽出することができる。

Garg ら [133] は、Wakahara らの手法と類似した手法を使い、カラーの雑誌画像からのテキスト、図、背景の分離を行った。この手法でも、まずカラー画像を、色空間でのクラスタリングによって、複数のレイヤーに分解する。その後、各黒画素連結領域がテキストか図か背景かを識別するために、**SVM** が使用される。上の手法と同じ理由により、この手法も、多色テロップから文字ストローク画像を正しく抽出することができる。

しかし、日本語の分離文字のように、1つの文字が複数の黒画素連結領域から構成される場合に、または、文字色と背景色が非常に類似している場合に、これらの2手法は、正しく文字ストロークを抽出することはできない。図 5.1 の a) は、カラーの原画像であり、b) は、色空間のクラスタリングで作成された色分解した1つのレイヤーの画像である。図を見てわかるように、日本語は、複数の部分で1文字を構成する場合があります。また、b) では、文字色と類似している背景部分から抽出された多数のノイズの黒画素連結領域が現れている。上の2手法で使われている **SVM** の学習の時に、学習サンプルの数には限界があり、かつ、全ての形状、サイズのノイズ黒画素連結領域を用意することは不可能であるので、**SVM** の識別能力には限界がある。そのため、上の2手法の文字ストローク抽出の精度にも限界がある。

本論文では、次のようなテロップをターゲットとする。

- 1) 1 つのテロップ文字列の中で複数の色が使用されているもの
- 2) 文字内にはテクスチャがなく、かつ、文字内のグラデーション変化は無い
- 3) 1 文字の中は、1 色
- 4) 移動するテロップではない静止テロップ

本論文では、複雑な背景上の 1 文字列中の多色テロップのための、新しい文字ストローク抽出方法を提案する。本手法は、多色テロップをよく使用する日本の TV 番組のビデオデータに適用することができる。本手法では、カラー画像を複数の色で分割し、複数のレイヤーを作成する。各レイヤーで、複数の黒画素連結領域を含む文字候補を抽出し、レイヤーを超えて、文字の確信度と文字の並び情報を使用した DP 手法により、最も妥当な文字候補の組み合わせを選択することで、文字ストローク画像を抽出する。

本論文では、第 5. 2 章で、提案するストローク画像抽出方法について記述し、第 5. 3 章で、提案方法を使用した評価実験結果を提示する。最後に、第 5. 4 章で結論と、将来の研究課題について記述する。

5.2 提案手法

図 5.2 に、提案手法の文字ストローク画像抽出を使用するテロップ認識処理の全体のフローチャートを示す。

ビデオデータが入力されると、まず、フレーム画像が抽出される。その後、フレーム画像から、[124] の手法によってテロップ領域が抽出される。次に、抽出されたカラーの領域画像から、提案手法により文字ストローク画像を抽出する。提案手法の出力は、文字だけの 2 値画像であり、この画像が OCR に入力され、認識結果のテロップのテキストが獲得される。

図 5.2 での「Color Caption Image(カラーテロップ画像)」から下の部分は、本提案ストローク抽出手法の詳細なフローチャートである。この詳細フローチャートの中の各部分を以下に示す。これらの処理では、多数のパラメータがあるが、これらは、第 5. 3 章で使われる画像とは別のパラメータ決定用の 10 程度の画像を使用した事前の実験によって予め決めた。

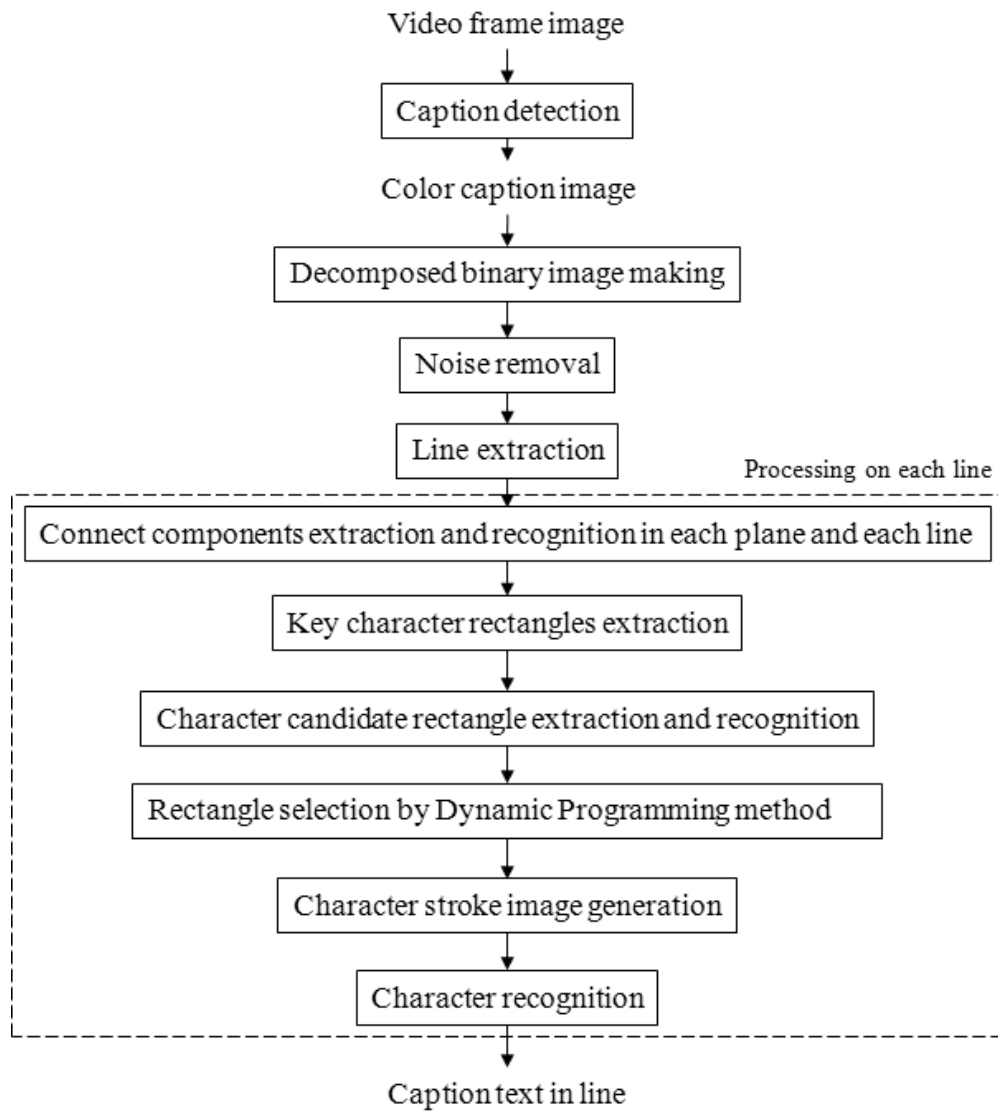


図 5.2 提案手法のフローチャート

5.2.1 色分解 2 値画像レイヤーの作成

図 5.3 a) で示されるカラーテロップ画像は、色空間上での ISO-DATA クラスタリング法によって、図 5.3 b) のように複数の 2 値画像のレイヤーに分解される。

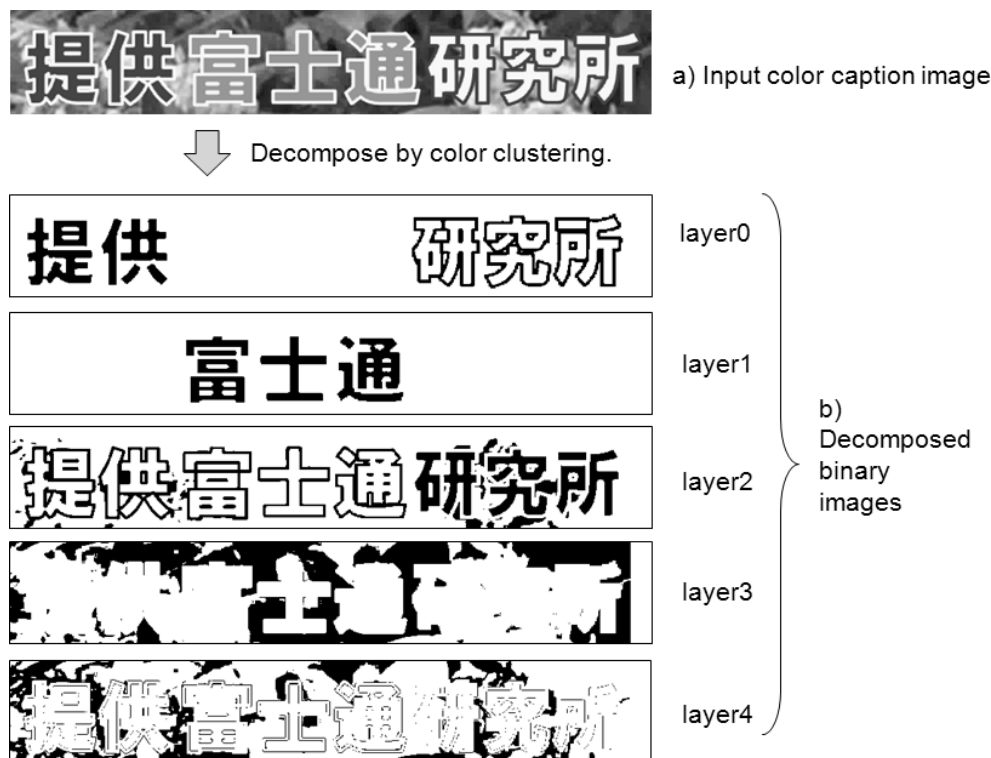


図 5.3 入力カラーテロップ画像と色分解レイヤーの各 2 値画像

5.2.2 使用する OCR エンジンと、OCR の確信度

今回使用する OCR エンジンでは、入力画像の輪郭の方向要素の数を使用した特徴ベクトルが使われる。文字が入力されると、入力画像から抽出した特徴ベクトルと、OCR 辞書中の特徴ベクトルとの間の距離が計算され、入力文字に最も近い辞書内の文字が認識結果として選択、出力される。また、入力画像が認識された文字であることの程度を示す「確信度」も出力される。確信度を計算するために、まず、認識結果の第 1 位候補文字と、第 2 位候補文字の距離値の逆数によって式 5.1 によるインデックス値 r が計算される。

$$r = \frac{v_1}{(v_1 + v_2)} \quad (5.1)$$

ここで、 $v_i=1/d_i$ 、また、 d_i は、OCR 結果の中で i 番目の候補の距離値を意味する。次に、事前に収集した学習用の約 22 万文字に対する評価によって、正しく認識された文字と、正しく認識できなかった文字に対する r 値のヒストグラムが、それぞれ求められる。

ここで、インデックス値 r について、正しく認識できた文字の頻度を $N_c(r)$ 、誤認識

した文字の頻度を $N_e(r)$ と定義する。確信度の値は、このヒストグラムを使用して計算される。日本語には類似文字が多いので、第1位候補の距離値と第2位候補の距離値は、ほとんど同じになる場合がある。この時、確信度は、第1位候補が正しい場合でも小さくなる。このような場合でも大きな確信度になるように、式5.2のような新しいインデックス値を導入した。

$$s = \frac{v_1}{(v_1 + v_j)} \quad (5.2)$$

ここで、 $v_j=1/d_j$ で、 d_j は、認識結果の中の第 j 番目の候補の距離値を表す。また、 j 番目の候補は次のように定義される。

まず、445518 文字の画像を使用して、事前予備試験によって類似文字テーブルを作成しておく。このテーブルでは、1つの入力キー文字に類似する文字がテーブルに記述されている。認識結果の第1候補文字をキー文字として、類似文字テーブルを参照すると、第1位候補文字に類似した文字の集合が獲得できる。認識結果の候補文字は、距離値によってソートされ小さい順に並んでいる。インデックス s の計算に使用される第 j 候補は、第2位候補から順に候補文字を参照していき、初めて類似文字集合に候補文字が無い時の順位として求められる。また、この候補の距離値を d_j とする。インデックス s を使用して、ダイナミックレンジが 0-1000 である確信度 c は式5.3によって計算される。式5.3で分かるように、第1位候補が正しい文字の場合に、確信度は大きな値をとることになる。

$$c = 1000 \frac{N_c(s)}{(N_c(s) + N_e(s))} \quad (5.3)$$

5.2.3 ノイズ除去

各レイヤーの2値画像において、ラベリングによって黒画素連結領域を抽出する。次に、黒画素連結領域は、OCR エンジンによって認識され、各黒画素連結領域の確信度が求められる。テロップ領域抽出の段階で、テロップ領域はテロップ文字に接する矩形よりもわずかに拡大してあるので、領域境界に接する黒画素連結領域は、ノイズとして削除する (図5.4)。

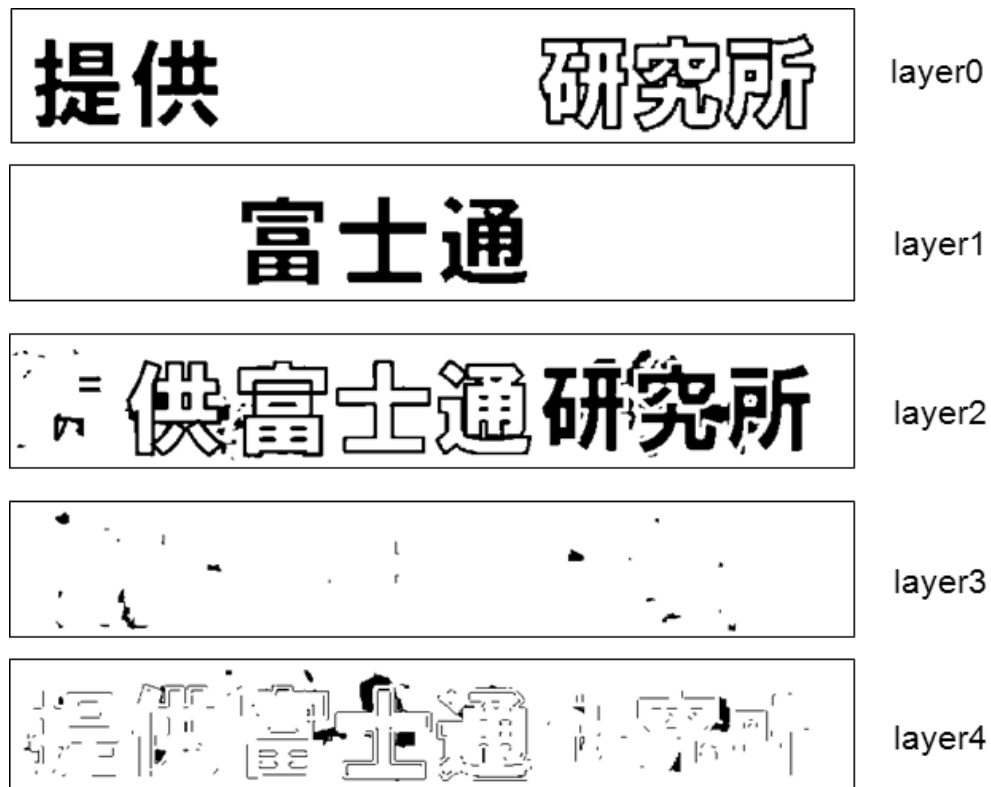


図 5.4 ノイズ除去後の各レイヤーの 2 値画像

5.2.4 文字列行領域抽出

まず、いくつかの文字がありそうなレイヤーを、黒画素数、および、領域内を左右方向の走査した場合の画素値の白黒変化数によって選択する。次に、選択されたレイヤーにおいて、確信度がしきい値より大きな黒画素連結領域を選択し、その外接矩形を求め、領域方向に沿って投影し、投影ヒストグラムを作成する。

文字列行領域は、投影ヒストグラム上でしきい値を使用することにより抽出される。ヒストグラムでしきい値を超える範囲が複数ある場合には、抽出されたテロップ領域に複数の文字列行があるとして、全てを行領域として抽出する。以降の処理は、各行領域の中で行われる。

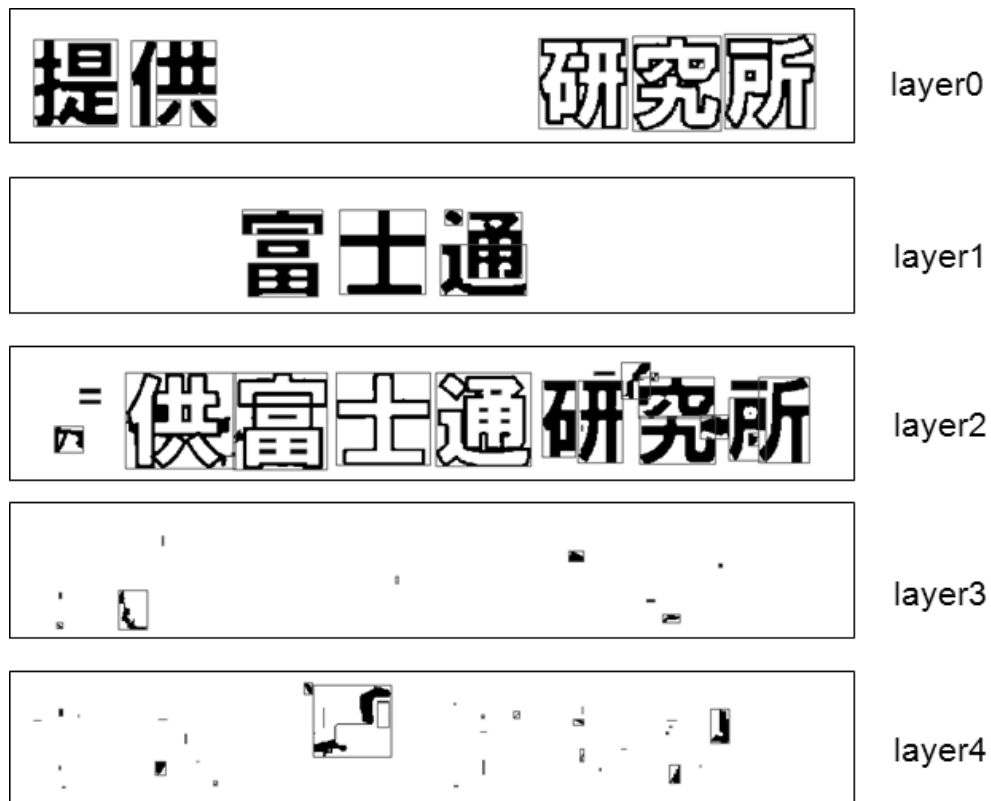


図 5.5 各レイヤーの黒画素連結領域

5.2.5 各レイヤー、各文字列行における黒画素連結領域抽出と文字認識

各レイヤーの抽出された文字列行領域で、黒画素連結領域がラベリングにより求められる（図 5.5）。

5.2.6 キー文字矩形抽出

1つの文字列行領域の中で、確信度がしきい値より大きな黒画素連結領域の外接矩形が「キー文字矩形」として抽出される（図 5.6）。

5.2.7 文字候補矩形抽出と認識

次に、キー文字矩形を拡大し、調査用ウィンドウを作成する。本手法では、この調査用ウィンドウを仮想的に各レイヤーの行方向に沿って移動させ、調査用ウィンドウが1つ以上の黒画素連結領域を完全に包含した場合に、その黒画素連結領域の組み合わせの外接矩形

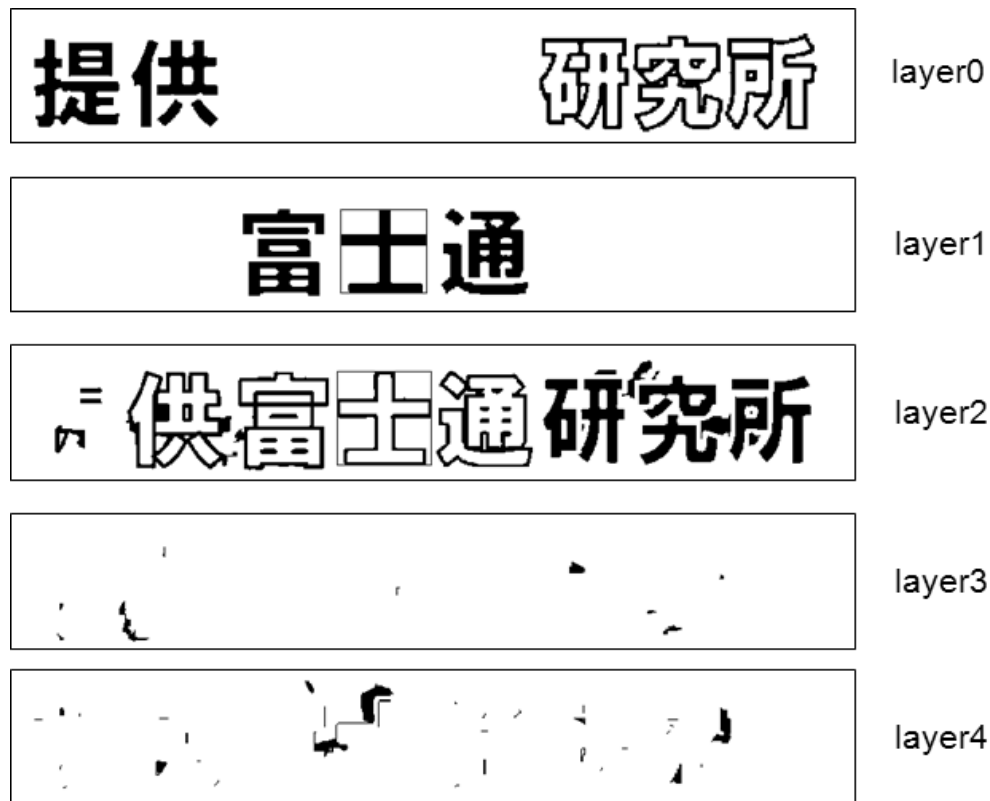


図 5.6 各レイヤーのキー文字矩形

形を文字候補矩形として抽出する。また、1つの文字候補矩形内の複数の黒画素連結領域の組み合わせでできる、黒画素連結領域の部分集合も別の文字候補矩形として抽出する(図 5.7)。

全ての文字候補矩形は、OCR エンジンによって文字認識され、認識結果文字と確信度が獲得される(図 5.8)。

5.2.8 ダイナミック・プログラミング手法による文字候補矩形の選択

1 行の領域において、多くの文字候補矩形が、複数のレイヤーにまたがって存在する。図 5.9 に示されるようにレイヤーをまたいで正しい文字矩形を選択するためにダイナミック・プログラミング (DP) 手法を使用する。

図 5.10 に示される様な、連続の 3 つの矩形によって定義される重み値の合計が計算され、中心の矩形にセットされる。

DP 手法では、行の先頭から現在の位置の矩形までの重みの合計値を矩形数で割った重

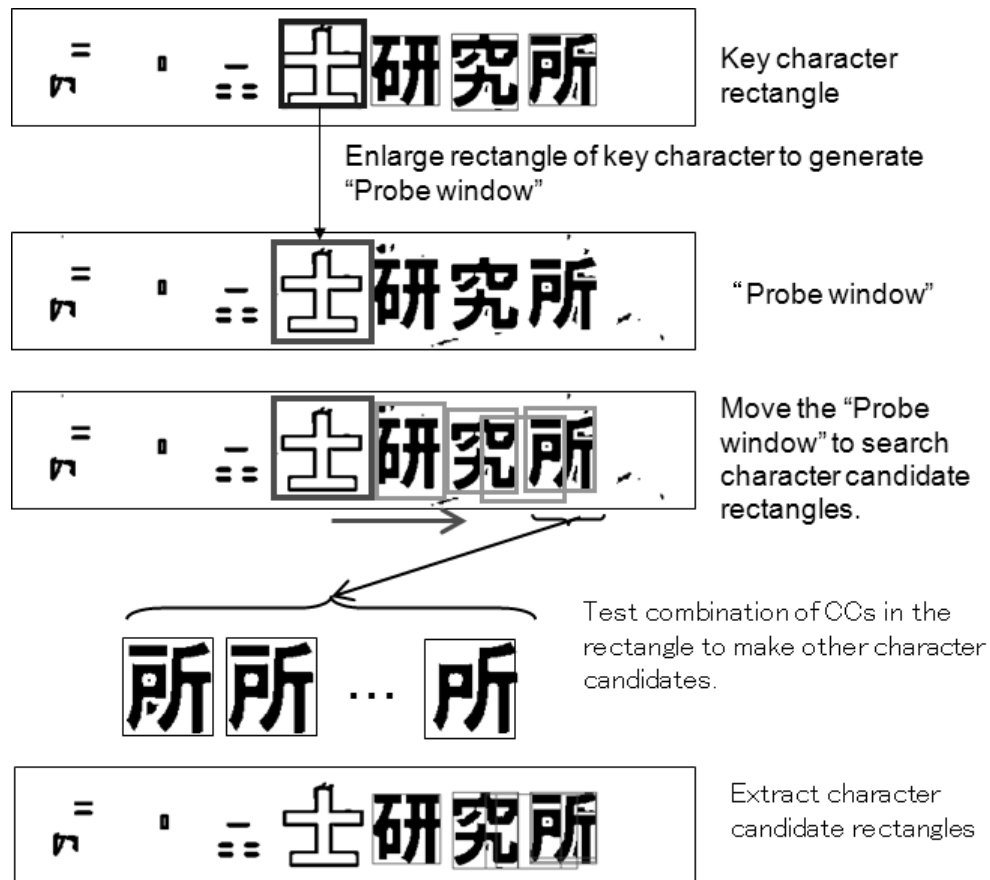


図 5.7 文字候補矩形抽出と文字認識

みの平均値が使用され、行末の矩形からみて、最も大きな重みの平均値となる矩形列が正しい文字矩形として選択される。ここで使用される重みは、以下のように定義される。

a) 確信度の重み

これは、中央の矩形の確信度 $CertaintyC$ と、中央の矩形の左にある矩形の重みの合計値である。式 5.3 に述べられているように、確信度のダイナミックレンジは 0-1000 である。この重みは、DP で使用されるすべての重みの中で、最も影響力のある重みである。

確信度の重み = 左の矩形の重み + $CertaintyC$

b) 矩形間距離の類似度の重み

この重みは、左の矩形と中央の矩形間の距離 $GapLC$ が、右の矩形と中央の矩形間の距離 $GapRC$ に近い場合に、この重みが大きくなるように、次の式で定義した。ここで、

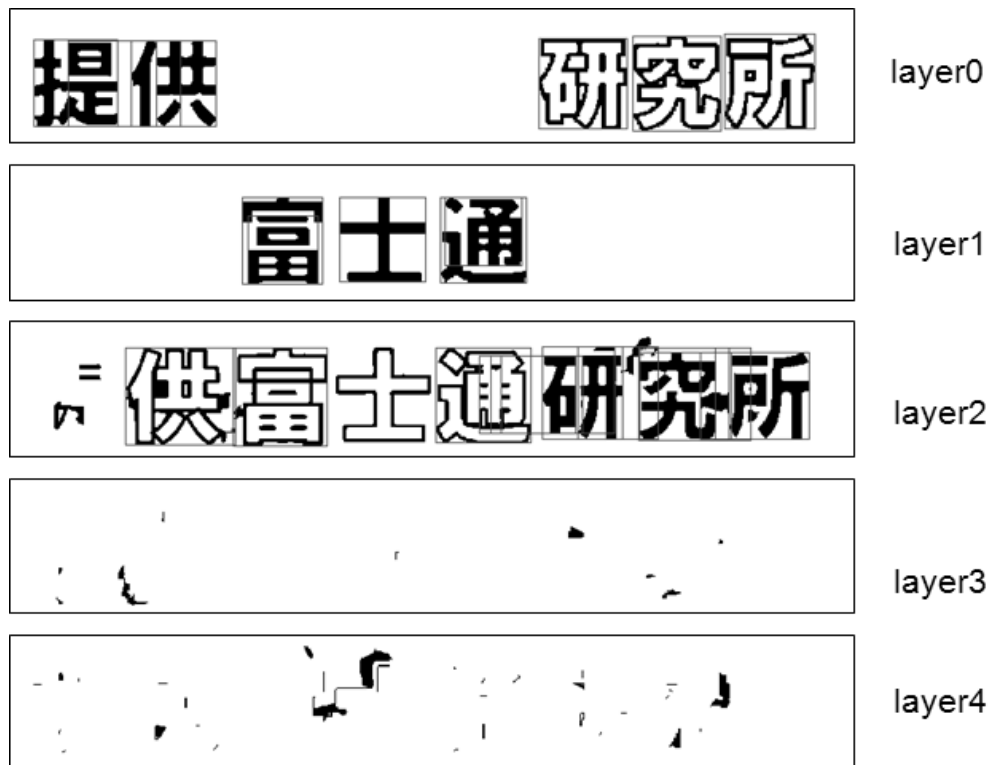


図 5.8 各レイヤーの文字候補矩形

W_d は 400 にセットされている。

$$W_d - |GapLC - GapRC|$$

c) 矩形端の類似性重み

3つの矩形の中の隣接した2つの矩形で、両方の矩形の上座標がお互いに類似している場合、この重みが大きくなるようにした。また、同様に、両方の矩形の下座標が類似している場合にも大きくなるように設計した。この重みは下の式で求められる。ここで、 W_p は 100 にセットされている。

$$(W_p - |Y_{minL} - Y_{minC}|) + (W_p - |Y_{maxL} - Y_{maxC}|) + \\ (W_p - |Y_{minR} - Y_{minC}|) + (W_p - |Y_{maxR} - Y_{maxC}|)$$

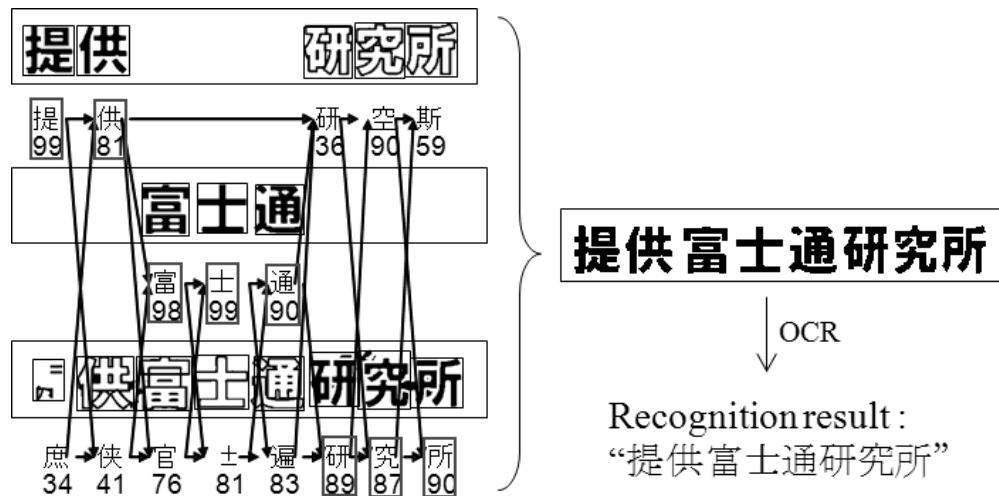


図 5.9 レイヤーをまたいだDP処理による文字ストローク画像抽出と文字認識

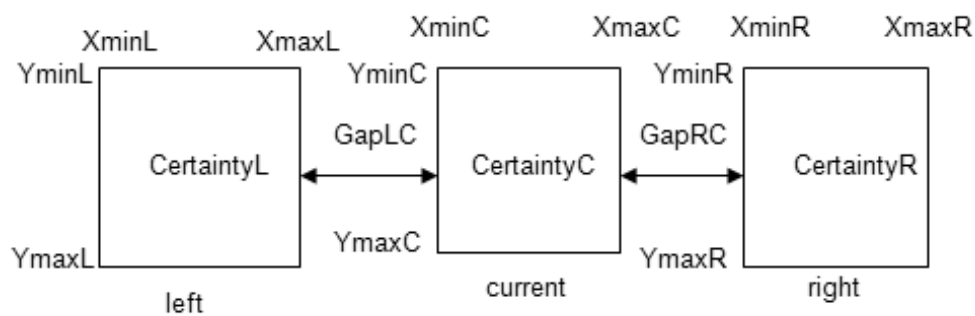


図 5.10 DP処理に使用される隣接3矩形の関係



図 5.11 選択された文字ストローク画像

5.2.9 文字ストローク画像生成

選択された矩形の中の文字画像を行領域の中に並べることによって、テロップ文字ストローク画像が抽出される（図 5.11）

表 5.1 テスト 1 の評価結果

手法	正しく認識された文字数	文字認識再現率
提案手法	405	76.0%
従来手法	312	58.5%

5.2.10 文字認識

抽出されたテロップ文字ストローク画像は、OCR エンジンによって認識され、テロップのテキストが獲得される。

5.3 評価実験と考察

提案手法の有効性を確認するために、2 種類の評価実験を行なった。テスト 1 は、複雑な背景上の 1 行内多色テロップだけを評価したものである。テスト 2 は、実際のテレビ番組のビデオ中のテロップ画像を評価対象としたものである。両方の評価実験において、評価は、認識された文字の中で正解の文字の数を正解テロップ文字数で割って定義される文字認識の再現率によって行なわれた。

また、提案手法と比較するために、従来方式として単純な手法を評価した。この手法は、[125] [126] [127] [128] [129] [130] で用いられているもので、1 行の文字列で 1 色のテロップだけを認識することができる単純な手法である。具体的には、この方法では、カラーテロップ画像は複数の 2 値画像レイヤーに分解され、図 5.4 に示されるように、ノイズが削除された後、1 つのレイヤーだけが文字ストローク画像として選択され、認識される。

テスト 1 では、テレビ番組ビデオなどから抽出された 50 の多色テロップを使用した。入力文字の合計文字数は 533 であった。結果を表 5.1 に示す。表 5.1 の結果から、提案手法は、比較した従来手法よりも高い再現率を得られることがわかる。

テスト 2 では、テレビ番組ビデオだけから抽出された 348 のテロップを使用した。このテストビデオに現れたテロップは、ほとんどすべてが複雑な背景上の 1 行文字列で、1 色テロップであった。合計文字数は 2547 であった。結果を、表 5.2 に示す。

表 5.2 の結果から、提案手法は、従来方式とほとんど同じ再現率となることがわかる。

表 5.2 テスト 2 の評価結果

手法	正しく認識された文字数	文字認識再現率
提案手法	1589	62.4%
従来手法	1595	62.6%

表 5.3 テスト 2 の評価結果

OCR エラー	色クラスタ リングエラー	DP 処理 エラー	文字候補矩形 抽出エラー	その他の エラー	合計
46	36	33	11	2	128

従って、提案手法は、1 色テロップに関して悪影響がないと判断できる。

本論文では、再現率が大きくなるように、各重みのバランスを調節した。結果として、確信度の重みが最も影響力のあるものになった。黒画素連結領域の外接矩形の形状に関する重みについて、ノイズの黒画素連結領域の矩形が正しい文字画像の矩形に似る場合があるので、単に形状と位置情報を使った重みだけでは、正しい矩形を選択するのは難しい。文字認識結果の確信度の重みは、このような場合に有力な情報となる。

また、文字矩形間の距離の均一性を示すピッチ情報は、すべての文字候補矩形集合の中から正しい矩形を選択する重要な情報であるので、矩形間距離の類似度の重みは、矩形端類似性重みより影響力がある。

次に、テスト 1 でのエラーの原因を分析した。エラーは、表 5.3 に記述されるように 4 つのタイプに分類した。

5.3.1 OCR エラー

通常の文書中の文字フォントと比較して、テロップの文字フォントが太いフォントである、または、特別な形状のフォントである場合、今回使用した OCR エンジン、文書画像認識のために設計されているので、このようなテロップ文字画像を正しく認識することができない。このタイプのエラーを少なくするためには、テロップで使われる様なより多くの文字フォント画像を使用して新たな OCR エンジンの辞書を作る必要がある。



図 5.12 色クラスタリングエラー

5.3.2 色クラスタリング・エラー

色空間でのクラスタリング処理では、同系色は、1 個のクラスタへグループ化される傾向がある。そのため、入力されたカラー画像の背景色と文字色のコントラストが小さい場合は、図 5.12 に示す様に、正しい文字ストロークを背景から分離することが難しくなる。この問題を解決するためには、レイヤー数を増加させるか、色クラスタリング処理の代わりに局所的な 2 値化手法の導入が必要になる。

5.3.3 DP エラー

上に記述されたように、確信度重みの影響は、矩形間距離の類似度の重み、および矩形端類似性重みより大きい。それだから、文字ストロークを含まないノイズ画像の文字候補矩形に正しい文字ストローク画像よりも大きな確信度が与えられた場合には、DP 処理は、文字として誤った矩形を選択してしまう傾向がある。この問題を解決するために、1 行の領域内で、文字線幅の同質性のような文字画像の他の特徴を導入する必要がある。さらに、DP 処理中の矩形間距離の類似度の重み、および矩形端類似性重みのような、1 行領域内の矩形の並びに関する重みも、高精度化のために調節する必要がある。

5.3.4 文字候補矩形抽出エラー

キー文字矩形が正しく抽出されないために、文字候補矩形が正しく抽出されない場合がある。キー文字矩形が正しく抽出されない理由は、以下のようなものがある。

- ・キー文字矩形抽出のためのしきい値より大きな確信度の矩形が無い
- ・キー文字矩形として抽出されたものが全て小さな矩形である。この場合、キー文字矩形の拡張によって作られる調査用ウィンドウは、正しい文字矩形を包含するのには小さすぎて、正しい文字矩形が抽出できない

このエラーを解決するためには、キー文字矩形を抽出する条件が緩め、正しいキー文字矩形が抽出されるようにする必要がある。

5.4 結論と今後について

本論文では、複雑な背景上の多色テロップのための認識手法を提案した。この手法では、カラーテロップ画像は、複数の2値画像のレイヤーに分解される。各レイヤーで、黒画素連結領域の認識確信度の使用により、黒画素連結領域の組み合わせで構成される文字候補矩形が生成される。文字矩形は、文字候補矩形の認識確信度と、矩形間距離の類似度および矩形端類似性の様な矩形の並び情報を使い、各レイヤーをまたいだ DP 処理によって選択される。

複雑な背景上の多色文字列の文字認識評価実験によって、1色テロップだけを認識することができる従来のテロップ認識手法より高精度な文字認識再現率を獲得できることを確認した。

また、今後は、提案手法と、1つの文字列中の多色文字列を認識することができる他の手法と比較し、さらに改良を加える予定である。さらに、テレビ番組の中で使用されるテロップ文字のために最適化された特別の文字フォント用の OCR 辞書を作り、高精度化を図るつもりである。また、より高い文字認識再現率のために、色分解の処理を改善する必要があり、局所的2値化手法の導入と、DP 処理の改良を検討しなければならない。

第 6 章

結論と今後の展望

本論文では、文字認識に関する研究の目標設定を行うために、先ず文字認識技術の歴史を紐解いた。文字認識技術は、1800 年代後半のスキャナ機能を持った機器の発明を起源として、人間の機械に文字を読ませたいという欲求の実現のために歴史を刻んできた。具体的な文字認識技術の実現は、1929 年の Gustav Tauschek の特許に始まる。その後、コンピュータの発明、発展と共に、様々な文字認識技術が研究、開発されてきた。また、最近ではインターネットの発展により文字認識技術の実現形態も大きく変化している。

本論文では、このような文字認識の歴史を振り返り、ICT 技術が発達した現在でもなお文字認識技術には課題が残っていて、特に実世界の文字をコンピュータに取り込む文字認識技術に大きな課題があることを再確認した。

これまでの文字認識技術の研究では、ハードウェアの制約をあまり考慮しないで高精度を目指した研究が主流であった。例えば、学習サンプル文字の分布を記述した辞書を使用した部分空間法、MQDF 認識処理、SVM 法、Boosting 法は、複雑な演算が必要であるため、高速処理が難しい課題がある。また、これらの手法は、辞書サイズが大きくなり、スマートフォンなどの小型端末で動作させることが難しく、また、並列に多数の処理を行う場合にハードウェア資源が大量に必要である。この様に、従来は、実用的な文字認識技術の研究では不十分であった。ここで、実用的な文字認識技術とは、高精度を維持したままの高速処理、かつ、コンピュータのメモリサイズの制限がある状況での高精度認識処理である。

本研究の目的は、従来の研究では不十分であった実用的な環境での高精度な文字認識技術について新しい手法を提案し、従来実現できなかった分野で文字認識技術を使う事により、実世界の文字をコンピュータに取り込み易くして、ICT によるサービスをさらに高度

化することにより人類の幸福に貢献することである。

実用的な文字認識を実現するための課題として、本論文では、高精度を維持したままの文字認識技術の高速化と省メモリ化に注目した。それぞれの課題は以下のようなものである。

- ・高速化の課題

日本語、中国語などのカテゴリの多い言語の文字では、従来の文字認識では、1つの入力文字パターンに対して、数千の辞書内カテゴリとの比較を行わなければならないため、計算量が大きく、処理時間が課題となっていた。

- ・省メモリ化の課題

従来の文字認識では、高精度認識を達成するために大きなメモリが使用できる環境での実行を前提としていた。しかし、近年、スマートフォンなどメモリ量が不十分なデバイスが出現し、高精度認識を実現するためのリソース不足が課題となっていた。また、クラウドなどで並列に多数の認識処理を行う場合にもハードウェア資源が大量に必要なため、個々の認識処理の必要メモリ量を小さくすることは課題である。

このように、近年では、大規模なコンピュータだけではなく、スマートフォンの様な小さな機器で文字認識技術が必要な場合がある。また、大規模なコンピュータでも、大量の画像を処理する必要があり、実用的な文字認識技術は必須の技術である。このような実用的な文字認識技術で課題となっているのは、認識精度をほとんど低下させないまま処理を高速化できる技術、および、高精度を維持したままの使用メモリ量の削減である。高精度な認識手法をベースとすると、必要なのは高速化、省メモリ化であり、以下の様な解決方法が知られている。

- ・高速化の従来の解決方法

高速化の課題を解決するために、よく用いられるアプローチとして、2段階選抜方式がある。すなわち、未知の文字が入力されたら、比較的軽い処理を用いて入力文字に類似した辞書内のカテゴリを選択する大分類を行う。その後、選択されたカテゴリだけを対象に、計算量は大きいですが、認識精度が高い手法を用いて詳細に認識を行い、最終的に入力文字に最も類似している辞書カテゴリ1つを認識結果として出力する詳細認識を行うもので

ある。

- ・省メモリ化の従来の解決方法

高精度を維持したままの省メモリ化の課題を解決するために、本論文では、先ず高精度認識が可能な **MQDF** 詳細認識に注目した。**MQDF** は、各カテゴリの分布状況を部分空間として辞書に持ち、それを使った認識を行うため、単純なユークリッド距離値を使用した認識処理よりは高精度な結果が得られる。しかし、メモリに展開される辞書サイズを考慮すると、ユークリッド距離値を使用した認識処理では、各カテゴリの平均値情報だけが辞書に必要なのに対し、**MQDF** 認識では、各カテゴリの平均値情報だけでなく、各カテゴリの分布状況を示す部分空間の情報が必要となるため、メモリを非常にたくさん使用していた。

この課題を解決するために **Compace MQDF** 法では、**MQDF** 辞書の中で大きな割合を占める各カテゴリの分布情報を示す特徴ベクトルを小さな部分ベクトルに分割し、それをクラスタリングして類似している部分ベクトルをまとめ、各クラスタにインデックス番号をつける。**MQDF** 辞書の分布情報のベクトル内容の代わりに、近似値として、部分ベクトル単位にインデックス番号で置き換えたものを使用することで、精度を低下させずに辞書サイズを削減している。しかし、それぞれの解決方法でも下の様に、まだ課題が残っている。

- ・高速化の残存課題

従来の2段階選抜方式の大分類処理では、大分類の性能についての基準が明確ではなかった。大分類の性能を表す基準として、大分類自体の処理速度、多くの辞書カテゴリから少数を選択する絞り込み能力、選択した少数のカテゴリに正解が入っている精度の3つが考えられる。従来の大分類処理では、これらの3つを同時に満たす手法がないという課題があった。

- ・省メモリ化の残存課題

MQDF 認識処理が高精度を実現できる理由は、特徴空間上の分布が類似している類似文字において、微妙な分布の違いを詳細に記述した分布情報を使用して入力文字と辞書カテゴリ間の評価値を計算するからである。しかし、**Compace MQDF** 法は、無条件で部分ベクトルをクラスタのインデックス番号に置き換えるので、この微妙な分布の違いを同じ物と見なしってしまう可能性が高い。その結果、類似文字間の認識能力が毀損され、オリジ

ナルの MQDF 辞書と比較した時に認識精度が低下していた。

このように、実用的な文字認識技術を実現するために、従来技術でも残存課題があった。本論文では、各課題の解決を図り、新しい手法を研究開発し、評価実験により有効性を確認した。具体的には、以下のような内容の研究を行った。

- ・高速化について

高精度を維持したままの高速化手段として、本論文では、2段階選抜方式を採用し、この中の大分類処理について改良を行った。まず、大分類が持つべき3つの要求項目を以下のように定義した。すなわち、

- 1) 非常に高い分類正解率
- 2) 大分類自身の高速性
- 3) 選択したカテゴリ数の小ささ

である。従来手法の連想整合法 (Associative Matching: AM 法) は、候補カテゴリを選択するためにハッシュテーブルとビット処理を使用することで、高速で高い効率を達成することができる。しかし、この手法では、学習サンプルを、ハッシュテーブルを構成する軸へ投影した時に、軸上の最小値と最大値だけの情報によりハッシュテーブルを構成し、学習サンプルの分布を考慮しないために、選択するカテゴリに冗長性があった。

本論文では、AM 手法をベースとして、各軸上で学習サンプルの分布を反映したハッシュテーブルとすることで、上に記述した3つの要求項目を同時に満足する新しい手法 Margin Added Hashing (MAH) 法を開発した。

また、本論文では、提案手法と、従来手法の AM 法、VQ クラスタリング法、ANN 法、および LSH 法について、116528 文字の日本語活字文字を使った評価実験を行い、性能を比較した。その結果、MAH 法は、大分類の分類精度、選択するカテゴリ数の小ささ、および総認識処理時間の短さのバランスにおいて従来手法よりもよい結果となることを示した。

- ・省メモリ化について

高精度を維持したまま、辞書サイズを縮小する手段として、本論文では、Long らの Compact MQDF 手法をベースとした研究を行った。

Compact MQDF 手法は、2 次識別関数で使用する辞書の中で固有ベクトルを小さい固定次元数の部分ベクトルに分割し、全ての部分ベクトルを対象にクラスタリングした結果のクラスタ番号で固有ベクトルを置き換えることで辞書サイズを削減する手法である。

Compact MQDF 手法は、オリジナルの通常の MQDF 手法と比べて、確かに辞書サイズは、大きく縮小できるが、文字認識精度がやや低下するという課題があった。

本論文では、Compact MQDF 手法をベースとして、Compact MQDF 手法とオリジナル MQDF 手法の中間の性能となる新しい Hybrid Compact MQDF 法を提案した。提案手法は、先ずオリジナルの MQDF 辞書の中の各カテゴリの代表ベクトルに対し、階層的クラスタリングを行い、一定距離値に収まる類似カテゴリのクラスタを作成する。この類似カテゴリのクラスタに所属するカテゴリは精密な認識が必要なために固有ベクトルはオリジナルのままとし、類似カテゴリのクラスタに所属しない他のカテゴリとは離れているカテゴリだけを対象に Compact MQDF の手法でオリジナル固有ベクトルを近似したものに置き換えて、小さいサイズの MQDF 辞書を作成する。この辞書を使用した Hybrid Compact MQDF 認識手法を評価したところ、Compact MQDF 手法の辞書サイズとオリジナル MQDF 手法の辞書サイズの中間の辞書サイズで、Compact MQDF 手法よりも高精度認識が実現できることを確認した。

・開発した認識技術の検証について

上の 2 つの改良を加えた文字認識技術を検証するために、本技術を映像中のテロップ認識へ適用した検証実験を行った。また、特にテロップの中で、1 行の中で文字の色が変化している場合について、テロップ文字を正しく読み取れるように、テロップ文字画像を色分解し、各色レイヤー上で文字領域を求め、最後に色レイヤーをまたいだ DP による最適な文字ストローク画像抽出と認識についての評価を行った。従来の 1 色テロップだけ読み取ることができるテロップ認識技術との比較実験で、開発した技術は、多色テロップ画像の読み取りでは、明らかな優位性があり、一色だけのテロップが混在した一般のテロップ画像での評価でも従来技術と同等の性能のまま維持されることがわかり、文字認識技術と、それを使った多色テロップ認識技術の有用性が確認できた。

このように、本論文では、2 段階選抜方式の大分類処理の改良により、高精度を保ったままの高速化を実現し、また、メモリ使用量が大きな 2 段階選抜方式の詳細認識処理の改良により、高精度を保ったままのメモリ使用量の削減を実現した。これにより、目標である、高精度を維持したままの高速認識処理、かつ、コンピュータのメモリサイズの制限がある状況での高精度認識処理を実現し、従来の研究では不十分だった実用的な文字認識技

術が実現できた。この技術を使用することにより、携帯端末などで実世界の文字をコンピュータに取り込み易くなり、ICTによるサービスをさらに高度化して、人類の幸福に貢献することができるようになる。

今後の研究の展望については、各課題について以下のようにまとめられる。

高速化手法について、提案した **Margin Added Hashing** 法では、各学習サンプルに加えるマージン値の選択が重要である。なぜならば、この値は、処理時間と分類精度の両方に影響を与えるからである。今後は、より実用的なアプリケーションを実現するために、各カテゴリの特徴空間の中での分布を考慮して、各学習サンプル毎に最適なマージン値を選択する調査を行っていく予定である。

また、今回開発した手法は、ハッシュ法をベースとしているため、学習データに現れない外れデータが入力された場合には、大分類結果の中に正解カテゴリを入れることができない。今後は、大分類の結果何も候補カテゴリが得られない場合に、距離を使用した類似カテゴリ抽出法や、類似カテゴリテーブルを使用して、大分類で抽出されたカテゴリに類似したカテゴリを追加する手法など、なるべく絞り込みカテゴリ数を増やさずに大分類の精度を高めるハイブリッド手法を検討していく予定である。

また、省メモリ化の手法については、今回は **MQDF** 辞書の中の固有ベクトルだけを対象にクラスタリングを行い、メモリ量を削減した。今後は、固有ベクトルだけでなく、固有値や代表ベクトルなども近似の対象として、さらに省メモリ化を進めることを検討している。また、クラスタリング結果の代表ベクトルで置き換える **lossy** な近似だけでなく、完全一致するベクトルを1つのクラスタで代表する **lossless** の圧縮処理との比較や、一部分に **lossless** 圧縮処理を使用することも行う予定である。

映像中のテロップ認識については、今後は、提案手法と、1つの文字列中の多色文字列を認識することができる他の手法と比較し、さらに改良を加える予定である。さらに、テレビ番組の中で使用されるテロップ文字のために最適化された特別の文字フォント用の **OCR** 辞書を作り、高精度化を図るつもりである。また、より高い文字認識再現率のために、色分解の処理を改善する必要があり、局所的2値化手法の導入と、**DP** 処理の改良を検討しなければならない。

さらに、今回の比較評価は、一部のものだけであったが、今後は、大分類としてニューラルネットワークや **Boosting** などを使用する場合との比較や、ニューラルネットワーク、**SVM**、**Boosting** などの高精度認識方式毎に省メモリ化を検討するなど、他のアプローチ

をベースとした改良、比較評価を行う必要がある。

また、今回作成された実用的な文字認識技術をカメラベースの認識システムへ適応して、評価する必要があるが、その場合には、認識部以外の前処理部分の研究、構築が必要である。ここで、前処理部とは、高速な文字列領域抽出、文字ストローク画像抽出などを指す。その上で、実際のスマートフォンなどに実装して、実験を行い、同じ認識技術をクラウドで実現する場合との比較なども行っていく予定である。

謝辞

本研究の全過程を通じ、懇切なるご指導とご鞭撻を賜りました東北大学情報科学研究科応用情報科学専攻 加藤 寧教授に心から感謝申し上げます。また、本研究をまとめるにあたって、貴重なお時間を割いて頂き、丁寧なるご教示を賜りました東北大学工学研究科通信工学専攻 大町真一郎教授に謹んで深謝致します。また、大学院博士後期課程において、親切なるご指導とご助言を賜りました東北大学情報科学研究科応用情報科学専攻 田中和之教授に厚く御礼申し上げます。

本研究は、株式会社富士通研究所において、社内の多くの方々のご指導とご協力を得て行ったものが元となっています。本研究の機会を与えていただくとともに、業務の傍らで本論文を纏めるにあたり、暖かいご配慮を賜った、株式会社富士通研究所元社長 村野和雄氏、元常務取締役（現株式会社アドバンテスト取締役）吉川誠一氏、元常務取締役（現早稲田大学大学院国際情報通信研究科教授）津田俊隆氏、元取締役（現富士通株式会社執行役員）上原三八氏、元言語・メディア研究部部長（現株式会社米国富士通研究所 **Senior vice president**）増本大器氏、株式会社富士通研究所社長 富田達夫氏、常務取締役 佐々木繁氏、取締役 吉澤英樹氏、メディア処理システム研究所所長 鈴木祥治氏、イメージコンピューティング研究部部長 清水雅芳氏、同主管研究員 上原祐介氏、同主管研究員（現フィールド・イノベーション本部フィールド・イノベータ育成推進室）堀田悦伸氏に心から御礼申し上げます。また、研究を進めるにあたり日々様々なご支援とご配慮を頂きました株式会社富士通研究所の研究部の先輩、同僚、後輩の方々に心から御礼申し上げます。最後に、いつも暖かく励ましてくれた、家族に感謝します。

2014 年 1 月

参考文献

- [1] A. C. Arcadi, “Vocal responsiveness in male wild chimpanzees: implications for the evolution of language,” *Journal of Human Evolution*, 39, pp. 205–223, 2000.
- [2] E. Klarreich, “Biography of Richard G. Klein,” in *Proc. of the National Academy of Sciences of the United States of America*, 101(16), pp. 5705–5807, 2004.
- [3] A. Vanka, “Skip Trie Matching for Real-Time OCR Output Error Correction on Smartphones,” Graduate Studies 5-1-2013, Utah State University, 2013. May 2005.
- [4] H. Schantz, “The History of OCR. Manchester Center,” VT: Recognition Technologies Users Association, 1982.
- [5] P. W. Handel, “Statistical machine,” U.S. Patent 1915993, 1933.
- [6] G. Tauschek, “Reading machine,” U.S. Patent 2026329, 1935.
- [7] Popular Science, “Reading Machine Speaks Out Loud,” *Popular Science*, 1949.
- [8] New York Times, “Washington Daily News, April 27, 1951,” *New York Times*, December 26, 1953.
- [9] D. H. Shepard, “Apparatus for reading,” US. Patent 2663758, 1953.
- [10] 日本自動認識システム協会「自動認識の歴史」. ホームページ. [Online]. Available: <http://www.jaisa.jp/about/pdfs/chronology.pdf>
- [11] M. H. Glauberman, “Character Recognition for business machines,” *Electronics*, pp. 132–136, 1956.
- [12] E. R. A., “An electronic reading automation,” *Electronic Engineering*, 29, pp. 189–190, 1957.
- [13] T. Iijima, Y. Okumura, and K. Kuwabara, “New process of character recognition using sieving method,” *Information and Control Research*, vol. 1, no. 1, pp. 30–35, 1963.
- [14] R. B. Johnson, “Indicia controlled record perforating machine,” US. Patent 2741312, 1956.

- [15] T. L. Dimond, "Devices for reading handwritten characters," Proc. of Eastern Joint Computer Conj., pp. 207–213, 1968.
- [16] L. A. Kamensky, "The simulation of three machines which read rows of handwritten Arabic numerals," IRE Trans. Electron. Comput., vol. EC-10, pp. 489–501, 1961.
- [17] B. A. Glucksman, "Classification of mixed-font alphabets by characteristic loci," Dig. 1st Ann. IEEE Comput. Conj., pp. 138–141, 1967.
- [18] R. K. Gerlach, "Wide-tolerance optical character recognition for existing printing mechanisms," in Optical Character Recognition, G. L. Fischer et al., Eds. McGregor & Wemer, pp. 93–114, 1962.
- [19] C. C. Heasley Jr., and G. L. Fischer Jr. "Some elements of optical scanning," Optical Character Recognition, G. L. Fischer et al., Eds. McGregor & Wemer, pp. 15–26, 1962.
- [20] J. J. Leimer, "Design factors in the development of an optical character recognition machine," Information Theory, IRE Transactions on Volume:8, Issue: 2, 1962.
- [21] E. C. Greanias, "Some important factors in the practical utilization of optical character readers," Optical Character Recognition, G. L. Fischer et al., Eds. McGregor & Wemer, pp. 129–146, 1962.
- [22] S. M. Elizabeth, "Research and Development in the Computer and Information Sciences," United States Bureau of Standards, Technical Report Archive and Image Library, pp. 108–108, 1970.
- [23] L. P. Honvitz, and G. L. Shelton, "Pattern recognition using autocorrelation," Proc. IRE, vol. 49, no. 1, pp. 175–185, 1961.
- [24] T. Iijima, "Basic theory on normalization of pattern," Bulletin of Electrotechnical Laboratory, vol. 26, no. 5, pp. 368–388, 1962.
- [25] T. Iijima, "Basic theory of feature extraction for visual pattern," T. R. on Automata and Automatic Control, IECE Japan, 1963.
- [26] D. Marr, and E. Hildreth, "Theory of edge detection," Proc. of Royal Society of London, Series B, Biological Sciences, vol. 207, No. 1167, pp. 187–217, 1980.
- [27] R. L. Cosgriff, "Identification of shape," Rep. 820-1 1, ASTIAAD 254 792, Ohio State Univ. Res. Foundation, Columbus, 1960.
- [28] R. W. Weeks, "Rotating raster character recognition system," AIEE Trans., vol. 80, pt. I, Communications and Electronics, pp. 353–359, 1961.

-
- [29] T. Sakai, M. Nagao, and Y. Shinmi, "A character recognition system," *Proc. of Annual Conj. IECE Japan*, pp. 450–450, 1963.
 - [30] E. C. Greanias, P. E. Meagher, R. J. Norman, and P. Essinger, "The recognition of handwritten numerals by contour analysis," *IBM J. Res. Develop.*, vol. 7, pp. 2–13, 1963.
 - [31] I. Sheinberg, "The INPUT2 document reader (a new optical character recognition system)," *Pattern Recognition*, vol. 2, pp. 167–178, 1970.
 - [32] H. Genchi, K. Mori, S. Watanabe, and S. Katsuragi, "Recognition of hand-printed numerical characters for automatic letter sorting," *Proc. of IEEE*, vol. 56, pp. 1292–1301, 1968.
 - [33] S. Watanabe, "Karhunen-Loeve expansion and factor analysis: theoretical remarks and applications," *Trans. 4th Prague Conf Info. Theory, Statis. Decision Funct., Random Process*, pp. 635–635, 1965.
 - [34] K. Fukunaga, "Introduction to Statistical Pattern Recognition," Academic Press, 1972.
 - [35] S. Watanabe, and N. Pakvasa, "Subspace method of pattern recognition," *Proc. 1st Int. J. Conf. on Pattern Recognition*, pp. 25–32, 1973.
 - [36] E. Oja, "Subspace Methods of Pattern Recognition," Research Studies Press, 1983.
 - [37] NEC (Nippon Electric Company), "Improvements in or relating to character recognition apparatus," U.K. Patent 1124130, 1968.
 - [38] J. H. Munson, "Experiments in the recognition of hand-printed text: Part I, Character recognition," *Proc. of AFIPS' 68*, vol. 33, pp. 1125–1138, 1968.
 - [39] M. Yasuda, "Research on character recognition systems based on correlation," Doctoral thesis, Tokyo University, 1981.
 - [40] Y. Yamashita, K. Higuchi, Y. Ydmada, and Y. Haga, "Classification of hand-printed Kanji characters by structural segment matching method," *Tech. Rep. IECE Japan*, PRL82-12, no. 12, pp. 25–25, 1982.
 - [41] H. Yamada, T. Saito, and K. Yamamoto, "Line density equalization - A nonlinear normalization for correlation method," *Trans. IECE Japan*, vol. J67D, no. 11, pp. 1379–1388, 1984.
 - [42] J. Tsukumo, and H. Tanaka, "Classification of hand-printed Chinese characters using nonlinear normalization and correlation methods," *Proc. 9th IJCPR*, 14-17, pp. 168–171, 1988.

- [43] 津雲, 浅井, “文字認識技術の最近の動向,” 電子情報通信学会研究報告, 画像工学研究会, Vol. IE88-5, No. 14, pp. 31–38, 1988.
- [44] 山本, “弛緩整合法による手書き教育漢字認識,” 電子情報通信学会, Vol. J65-D, No. 9, pp. 1167–1174, 1982.
- [45] 梅田, “手書き漢字認識手法の個性的なサーベイ,” 「手書き文字認識技術の過去・現在・未来」シンポジウム公演論文集, pp. 54–63, 1994.
- [46] 山田, “輪郭D P 整合法と手書き漢字認識への適用,” 電子情報通信学会, Vol. J67-D, No.3, pp. 351–358, 1984.
- [47] 高橋, 内藤, 増田, “外郭構造情報を利用したストローク抽出法による手書き漢字認識,” 電子情報通信学会, Vol. J67-D, No.9, pp. 1052–1059, 1984.
- [48] 田中, 青田, 塩野, 真田, 手塚, “手書き漢字の部分パターン抽出の一手法,” 電子情報通信学会, Vol. J69-D, No.6, pp. 911–920, 1986.
- [49] 目黒, 梅田, “マルチフォント印刷漢字認識装置,” 電子情報通信学会, Vol. J67-D, No.8, pp. 908–915, 1984.
- [50] 山本, 中島, 中田, “階層的パターンマッチングによる漢字認識の基礎 —印刷漢字認識の研究—,” 電子情報通信学会, Vol. J56-D, No.12, pp. 714–721, 1973.
- [51] 岡, “図形からのセル特徴の抽出,” 電子情報通信学会, Vol. J65-D, No.10, pp. 1219–1226, 1982.
- [52] 岡, “セル特徴を用いた手書き漢字の認識,” 電子情報通信学会, Vol. J66-D, No.1, pp. 17–24, 1982.
- [53] 鶴岡, 栗田, 原田, 木村, 三宅, “加重方向ヒストグラム法による手書き漢字ひらがな認識,” 電子情報通信学会論文誌, Vol. J70-D, No.7, pp. 1390–1397, 1987.
- [54] 本郷, 木内, 岸, “日本語ドキュメントプロセッサ XP-70S,” 富士時報, Vol.64, No.8, 1991.
- [55] 松村, “卓上型手書き漢字 OCR;CLL-2000,” 三洋電機技法, Vol. 18, No.1, pp. 3–12, 1986.
- [56] S. V. Rice, F. R. Jenkins, and T. A. Nartker, “The Fifth Annual Test of OCR Accuracy,” Information Science Research Institute, 1996.
- [57] R. Holley, “How Good Can It Get? Analysing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs,” D-Lib Magazine. Vol. 15, 3/4, 2009.
- [58] C. Y. Suen, R. Plamondon, A. Tappert, A. Thomassen, J. R. Ward, and K. Yamamoto,

-
- “Future Challenges in Handwriting and Computer Applications,” 3rd International Symposium on Handwriting and Computer Applications, Montreal, 1987.
- [59] 森, 澤木, “低品質文字の認識手法とその応用に関するサーベイ,” 電子情報通信学会技術研究報告, PRMU2001-275, No.713, pp. 25–40, 2001.
- [60] 黄瀬, 大町, 内田, 岩村, “カメラを用いた文字認識・文書画像解析の現状と課題,” 信学技報 PRMU, 104(742), pp. 85–90, 2005.
- [61] C. Cortes, and V. Vapnik, “Support Vector Networks,” *Machine Learning*, Vol. 20, pp. 273–297, 1995.
- [62] Y. Freund, and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Proc. of the Second European Conference on Computational Learning Theory*, pp. 23–37, 1995.
- [63] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep Big Simple Neural Nets For Handwritten Digit Recognition,” *Neural Computation* Vol. 22, pp. 3207–3220, 2010.
- [64] H. S. Park, and S. W. Lee, “A Truly 2-D Hidden Markov Model for Off-line Handwritten Character Recognition,” *Pattern Recognition*, vol. 31, no.12, pp. 1849–1864, 1998.
- [65] D. Doermann, J. Liang, and H. Li, “Progress in camerabased document image analysis,” *Proc. of ICDAR’03*, pp. 606–616, 2003.
- [66] 池谷, 中島, 佐藤, 池田, 神原, 横矢, 山田, “紙面を対象としたカメラパス推定によるビデオモザイクングおよび超解像,” 信学技報, PRMU2003-223, 2004.
- [67] C. Wu, and G. Agam, “Document image de-warping for text/graphics recognition,” *Lecture Notes in Computer Science (Joint IAPR International Workshops SSPR 2002 and SPR 2002)*, Vol. 2396, pp. 348–357, 2002.
- [68] T. Yamazoe, M. Etoh, T. Yoshimura, K. Tsujino, “Hypothesis Presentation Approach to Scene Text Recognition with Weighted Finite-State Transducer,” *Proc. of ICDAR2011*, pp. 359–363, 2011.
- [69] 山田, 仙田, “携帯カメラを用いたユビキタス情報インタフェース,” 情報処理, 45, 9, pp. 923–927, 2004.
- [70] K. Wang, J. A. Kangas, “Character location in scene images from digital camera,” *Pattern Recognition*, vol.36, 10, pp. 2287–2299, 2003.
- [71] Y. Kusachi, A. Suzuki, N. Ito, and K. Arakawa, “Kanji recognition in scene images

- without detection of text fields robust against variation of viewpoint, contrast, and background texture,” Proc. of ICPR’04, pp. 23–26, 2004.
- [72] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, “Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning,” Proc. of ICDAR2011, pp. 440–445, 2011.
- [73] 勝山, 桂, 皆川, 堀田, “複雑背景中の多色テロップの認識方式の検討,” 信学技報, vol. 110, no. 467, PRMU2010-247, pp. 61–66, 2011.
- [74] 堀, 三田, “テロップ認識のための映像からのロバストな文字郡抽出法,” 信学論 (D-II), vol. J84-D-II, no. 8, pp. 1800–1808, 2001.
- [75] 澤木, 村瀬, 萩由, “劣化推定に基づいた辞書の自動選択による本棚画像中の文字認識,” 映情学誌, vol. 54, no. 6, pp. 881–886, 2000.
- [76] S. Omachi, F. Sun, and H. Aso, “A Noise adaptive Discriminant Function and Its Application to Blurred Machine-printed Kanji Recognition,” IEEE Trans. PAMI, vol. 22, no. 3, pp. 314–319, 2000.
- [77] 長谷, 米田, 酒井, 丸山, “カラー文書画像中の文字領域抽出を目的とした色分割についての検討,” 信学論 (D-II), vol. J83-D-II, no. 5, pp. 1294–1304, 2000.
- [78] T. Perroud, K. Sobottka, L. Hall, and H. Bunke, “Text Extraction from Color Documents-Clustering approach in Three and Four Dimensions,” Proc. of 6th ICDAR, pp. 937–941, 2001.
- [79] 濱中, “主要色線分法を用いた低解像度カラー画像からの文字抽出・認識手法,” 信学技報, PRMU2001-191, 2001.
- [80] 内田, 迫江, “カテゴリ固有変形の線形結合モデルに基づく弾性マッチング法,” 電子情報通信学会論文誌 (D-II), vol. J87-DII, no. 2, pp. 639–648, 2004.
- [81] M. Radmehr, S. M. Anisheh, M. Nikpour, and A. Yaseri, “Designing an Offline Method for Signature Recognition,” World Applied Sciences Journal vol. 13, (3), pp. 438–443, 2011.
- [82] M. Liwicki, M. I. Malik, C. E. van den Heuvel, X. Chen, C. Berger, R. Stoel, M. Blumenstein, and B. Found, “Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011),” Proc. of ICDAR2011, pp. 1480–1484, 2011.
- [83] 堀田, 武部, 直井, “文字特徴からの特徴合成に基づく単語一括認識,” 信学技報, PRMU99-239, 2000.

-
- [84] 徳本, 鈴木, 加藤, 根元, “候補あて名の優先度付けによる高速大分類法を用いた手書きあて名認識システム,” 信学論 (D-II), vol.J84-D-II, no.1, pp. 83–92, 2001.
 - [85] R. Palacios, A. Gupta, and P. S. Wang, “Handwritten bank check recognition of courtesy amounts,” *International Journal of Image and Graphics* Vol. 4, No. 2, pp. 1–20, 2004.
 - [86] A. Fischer, E. Indermuhle, V. Frinken, and H. Bunke, “HMM-Based Alignment of Inaccurate Transcriptions for Historical Documents,” *Proc. of ICDAR’2011*, pp. 53–57, 2011.
 - [87] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, “Historical Document Layout Analysis Competition,” *Proc. of ICDAR2011*, pp. 1516–1520, 2011.
 - [88] J. B. Baker, A. P. Sexton, V. Sorge, and M. Suzuki, “Comparing Approaches to Mathematical Document Analysis from PDF,” *Proc. of ICDAR2011*, pp. 463–467, 2011.
 - [89] D. Bainbridge, and T. Bell, “The Challenge of Optical Music Recognition,” *Computers and the Humanities*, vol.35, pp. 95–121, 2001.
 - [90] R. Jain, V. Frinken, C. V. Jawahar, and R. Manmatha, “BLSTM Neural Network Based Word Retrieval for Hindi Documents,” *Proc. of ICDAR’2011*, pp. 83–87, 2011.
 - [91] H. Wei, G. Gao, and Y. Bao, “A Method for Removing Inflectional Suffixes in Word Spotting of Mongolian Kanjur,” *Proc. of ICDAR’2011*, pp. 88–92, 2011.
 - [92] V. Margner, and H. El Abed, “ICDAR 2011 - Arabic Handwriting Recognition Competition,” *Proc. of ICDAR2011*, pp. 1444–1448, 2011.
 - [93] T. Long, and L. Jin, “Building compact MQDF classifier for large character set recognition by subspace distribution sharing,” *Pattern Recognition*, vol.41, pp. 2916–2925, 2008.
 - [94] Y. Wang, and Q. Huo, “Building Compact Recognizers of Handwritten Chinese Characters Using Precision Constrained Gaussian Model, Minimum Classification Error Training and Parameter Compression,” *International Journal on Document Analysis and Recognition (IJDAR)*, Volume 14, Issue 3, pp. 255–262, 2011.
 - [95] M. Zhang, R. Alhajj, and J. Rokne, “Effectiveness of optimal incremental multi-step nearest neighbor search,” *Expert Systems with Applications*, Volume 37, Issue 8, pp. 6018–6027, 2010.
 - [96] Y. C. Liaw, M. L. Leou, and C. M. Wu, “Fast exact k nearest neighbors search using an orthogonal search tree,” *Pattern Recognition*, Volume 43, Issue 6, pp. 2351–2358, 2010.

- [97] M. Blachnik, and W. Duch, “Improving accuracy of LVQ algorithm by instance weighting,” *Lecture Notes in Computer Science* Vol. 6353, pp. 256–265, 2010.
- [98] M. Barrena, E. Jurado, P. M. Neila, and C. Pachon, “A flexible framework to ease nearest neighbor search in multidimensional data spaces,” *Proc. of Data & Knowledge Engineering*, Volume 69, Issue 1, pp. 116–136, 2010.
- [99] Y. S. Chen, Y. P. Hung, T. F. Yen, and C. S. Fuh, “Fast and versatile algorithm for nearest neighbor search based on a lower bound tree,” *Pattern Recognition*, Volume 40, Issue 2, pp. 360–375, 2007.
- [100] J. Toyama, M. Kudo, and H. Imai, “Probably Correct k-Nearest Neighbor Search,” *Pattern Recognition*, Volume 43, Issue 4, pp. 1361–1372, 2010.
- [101] J. Kybic, and I. Vnucko, “Approximate all nearest neighbor search for high dimensional entropy estimation for image registration,” *Proc. of Signal Processing*, Volume 92, Issue 5, pp. 1302–1316, 2012.
- [102] H. Jung, Y. D. Chung, and L. Liu, “Processing generalized k-nearest neighbor queries on a wireless broadcast stream,” *Information Sciences*, Volume 188, pp. 64–79, 2012.
- [103] K. K. Reddy, J. Liu, and M. Shah, “Incremental Action Recognition Using Feature-Tree,” *Proc. of the 12th International Conference on Computer Vision (ICCV2009)*, pp. 1010–1017, 2009.
- [104] T. Liu, A. Moore, A. Gray, and K. Yang, “An Investigation of Practical Approximate Nearest Neighbor Algorithms,” *Proc. of Neural Information Processing Systems (NIPS 2004)*, pp. 825–832, 2004.
- [105] Y. L. Qiao, Z. M. Lu, J. S. Pan, and S. H. Sun, “Fast k-nearest neighbor search algorithm based on pyramid structure of wavelet transform and its application to texture classification,” *Proc. of Digital Signal Processing*, Volume 20, Issue 3, pp. 837–845, 2010.
- [106] M. Muja, and D. G. Lowe, “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration,” *Proc. of International Conference on Computer Vision Theory and Applications (VISAPP’09)*, pp. 331–340, 2009.
- [107] D. M. Mount, and S. Arya, “ANN: A Library for Approximate Nearest Neighbor Searching,” <http://www.cs.umd.edu/mount/ANN/>
- [108] P. Indyk, and R. Motwani, “Approximate nearest neighbors: towards removing the

-
- curse of dimensionality,” STOC, pp. 604–613, 1998.
- [109] A. Gionis, P. Indyk, and R. Motwani, “Similarity Search in High Dimensions via Hashing,” Proc. of the 25th International Conference on Very Large Data Bases, pp. 518–529, 1999.
 - [110] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-Sensitive Hashing Scheme Based on p-Stable Distributions,” Proc. of the 20th annual symposium on Computational Geometry, pp. 253–262, 2004.
 - [111] A. Andoni, and P. Indyk, “Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions,” Proc. of the Symposium on Foundations of Computer Science (FOCS’06), pp. 1–10, 2006.
 - [112] M. Slaney, and M. Casey, “Locality-Sensitive Hashing for Finding Nearest Neighbors,” IEEE Signal Processing Magazine, Volume 25, Issue 2, pp. 128–131, 2008.
 - [113] K. Tanaka, and E. Kondo, “A scalable algorithm for monte carlo localization using an incremental E2LSH-database of high dimensional features,” Proc. of International Conference on Robotics and Automation (ICRA 2008), pp. 2784–2791, 2008.
 - [114] M. Toda, Y. Magome, and T. Kubota, “A high-speed rough classification method based on associative matching,” Systems and Computers in Japan, Vol. 30, Issue 9, pp. 34–43, 1999.
 - [115] Y. Katsuyama, A. Minagawa, Y. Hotta, S. Omachi, and N. Kato, “A New Pre-classification Method based on Associative Matching Method,” Proc. of DRR’2010. Vol. 7534SPIE, pp. 1–10, 2010.
 - [116] N. Kato, M. Suzuki, S. Omachi, H. Aso, and Y. Nemoto, “A Handwritten character Recognition System Using Directional Element Feature and Asymmetric Mahalanobis Distance,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 3, pp. 258–262, 1999.
 - [117] F. Sun, S. Omachi, and H. Aso, “An Algorithm for Constructing a Multi-Template Dictionary for Character Recognition Considering Distribution of Feature Vectors,” Proc. of the 14th International Conference on Pattern Recognition (ICPR’98), vol. 2, pp. 1114–1116, 1998.
 - [118] A. Ahmad, and L. Dey, “A k-mean clustering algorithm for mixed numeric and categorical data,” Proc. of Data & Knowledge Engineering, Volume 63, Issue 2, pp. 503–527,

- 2007.
- [119] C. L. Liu, R. Mine, and M. Koga, "Building compact classifier for large character set recognition using discriminative feature extraction," Proc. of the 8th ICDAR, pp. 846–850, 2005.
- [120] 織田、朱、小沼、徳野、未代、中川, "オフライン識別器を統合したオンライン手書き文字文字識別器の小型化," 電子情報通信学会論文誌 D-II, J90-D(9), pp. 2583–2594, 2007.
- [121] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," IEEE Trans. Pattern Anal. Mach. Intell. 9 (1), pp. 149–153, 1987.
- [122] 安藤, 尾関, "誤認識関数を最小化するクラスタリングアルゴリズム," 電子情報通信学会論文誌, A, Vol.J74-A, No.3, pp. 360–367, 1991.
- [123] 神嶌, "データマイニング分野のクラスタリング手法 (1) - クラスタリングを使ってみよう ! -, " 人工知能学会誌, vol.18, no.1, pp. 59–65, 2003.
- [124] T. Gui, J. Sun, S. Naoi, Y. Katsuyama, A. Minagawa, and Y. Hotta, "A Fast Caption Detection Method for Low Quality Video Images," 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 302–306, 2012.
- [125] T. Mita, and O. Hori, "Improvement of Video Text Recognition by Character Selection," Proc. of ICDAR2001, pp. 1089–1093, 2001.
- [126] F. Chang, G. C. Chen, C. C. Lin, and W. H. Lin, "Caption Analysis and Recognition For Building Video Indexing Systems," Multimedia Systems, Volume 10, Number 4, pp. 344–355, 2005.
- [127] T. Sato, T. kanade, E. K. Hughes, M. A. Smith, and S. Satoh, "Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions," Multimedia Systems, Vol.7, No. 5, pp. 385–395, 1999.
- [128] X. Tang, X. Gao, J. Liu, and H. Zhang, "Spatial-Temporal Approach for Video Caption Detection and Recognition," IEEE Transactions on Neural Networks, Vol.13, No. 4, pp. 961–971, 2002.
- [129] M. R. Lyu, and J. Song, "A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction," IEEE Trans. On Circuits and Systems for Video Technology, Vol.15, No. 2, pp. 243–255, 2005.

-
- [130] S. Antani, D. Crandall, and R. Kasturi, “Robust Extraction of Text in Video,” Proc. of 15th Int. Conf. Pattern Recognition, vol. 1, pp. 831–834, 2000.
 - [131] X. S. Hua, P. Yin, and H. J. Zhang, “Efficient Video Text Recognition using Multiple Frame Integration,” International Conference on Image Processing, Vol.2, pp. II397–II400, 2002.
 - [132] T. Wakahara, and K. Kita, “Binarization of Color Character Strings in Scene Images Using K-means Clustering and Support Vector Machines,” International Conference on Document Analysis and Recognition (ICDAR2011), pp. 274–278, 2011.
 - [133] R. Garg, E. Hassan, S. Chaudhury, and M. Gopal, “A CRF Based Scheme for Overlapping Multi-colored Text Graphics Separation,” International Conference on Document Analysis and Recognition (ICDAR2011), pp. 1215–1219, 2011.

発表文献

学会論文

- [1] Yutaka Katsuyama, Yoshinobu Hotta, Masako Omachi, and Shinichiro Omachi, “High Speed and High Accuracy Pre-Classification Method for OCR: Margin Added Hashing,” *IEICE TRANSACTIONS on Information and Systems*, vol. E96-D, no. 9, pp. 2087–2095, Sep. 2013.

国際会議発表論文（査読あり）

- [1] Yutaka Katsuyama, Akihiro Minagawa, Yoshinobu Hotta, Jun Sun, and Shinichiro Omachi, “A Study on Caption Recognition for Multi-color Characters on Complex Background,” in *2012 IEEE International Symposium on Multimedia (ISM)*, Dec. 2012, pp. 401–407.
- [1] Yutaka Katsuyama, Akihiro Minagawa, Yoshinobu Hotta, Shinichiro Omachi, and Nei Kato, “A New Pre-classification Method based on Associative Matching Method,” in *Proc. of SPIE-IS&T Electronic Imaging, Document Recognition and Retrieval XVII*, SPIE Vol. 7534, Jan. 2010, pp. 1–10.